SEST Pegasus administration handbook

Last Update: 26th of November 2000

Mikael Lerner

Contents

Chapter 1

Pegasus directory structure

The Pegasus system as well as software for data reduction and all the home directories for observerations and Pegasus development reside on a disk partition called `/aux2`. There are also two other disk partitions being used for observations: `/archive` which contains the on-line archive of completed observing projects, and `/logs` which contains various log–files (for example, the AOS drift logs). The `/cdscr` partition is used for backing up data to CDs. Various files on `/tmp` are also important for the Pegasus system.

Several of these partitions also exists on other machines: the `/aux2` partition also exists on CARINA, as well as the `/logs` partition, although the latter one does not have the same contents as its VELA counterpart. The `/archive` also exist on PUPPIS although it is called `/vela_archive` instead.

All partitions also exists on the emergency disk of VELA. This disk is updated on a bi-weekly basis and is only supposed to be used in case of a crash of the main external disk on VELA or if VELA crashes and CARINA has to take over its job. The disk contains mirror copies of the root partition and the `/aux2` partition, while the `/archive`, `/logs` and `/cdscr` partitions only are installed in skeleton versions in order to save disk space.

1.1 Disk usage

1.1.1 `/aux2`

There is one `/aux2` partition on VELA and another one on CARINA. The latter one is supposed to be a mirror of the one on VELA that can be used for software development.

There is also a mirror partition of `/aux2` on the emergency disk on VELA. Since the emergency disk is smaller (only 4 GB), the `/aux2` partition has been reduced to 1.5 GB, which should be enough since the typical usage of `/aux2` is around 1 GB.

1.1.2 /archive

There is one `/archive` partition on VELA and another one that is called `/vela_archive` on PUPPIS. They both contain the on-line observing data archive and are supposed to have identical content. Both areas have a size of 4 GB.

There is also a mirror partition of `/archive` on the emergency disk on VELA. Since the emergency disk is smaller (only 4 GB), the `/aux2` partition has been reduced to 256 MB and nothing is copied into it during the bi-weekly updates. This should be enough to store the archives for a couple of months.

1.1.3 /logs

The `/logs` partition on VELA is a 512 MB area for various kinds of lg files generated by the Pegasus system. There is also a similar partition on CARINA with the size of 1 GB. This partition is used to store daily log files of the Pegasus logs. This is only done on CARINA.

The mirror partition of `/logs` on the emergency disk contains the proper subdirectory structure but contains only empty log files in order to save disk space. The partition is assigned 224 MB of disk space which should be enough for many months of usage.

1.1.4 /cdscr

The `/cdscr` partition on VELA is a 1.5 GB scratch area that is only to be used for the temporary storage of a CD image when backing up data onto a CD. It is also used by some system administration programs run on VELA.

The mirror partition of `/cdscr` on the emergency disk is just an empty file system with a size of 24 MB, and can thus not be used for anything.

1.1.5 /tmp

The `/tmp` area is not a separate partition, but is just a part of the root partition, which is a ...

1.1.6 Updating external disks

1.1.7 Maintaining the on-line archives

The on-line archives of observational data is basically maintaining itself. For a detailed description of them, see the section on "*On-line data archives*" in the **SEST system administration handbook**.

1.2 General directory structure

This section contains an overview of the directory structure of the partitions used by Pegasus. A more detailed description of the structure of the Pegasus system itself will be given in the section called "*Directory structure of* `/aux2/local/cfht`".

1.2.1 `/aux2`

The `/aux2` partition is owned by `root:root` and contains the Pegasus system, all the home directories for the observing accounts and software needed for the Pegasus system like data reduction programs.

There are 13 subdirectories on `/aux2` with the following ownerships and permissions:

Name	**Owner**	**Permission**	**Subdirectories**	
			One level	**Nested**
`Drp`	`root:sys`	`rwxr-xr-x`	yes	yes
`TT_DB`	`root:root`	`rwxr-xr-x`	no	no
`bin`	`root:sys`	`rwxr-xr-x`	no	no
`catalog`	`root:sys`	`rwxr-xr-x`	no	no
`dvips`	`root:sys`	`rwxr-xr-x`	yes	yes
`gag`	`root:sys`	`rwxr-xr-x`	yes	yes
`gcc`	`root:sys`	`rwxr-xr-x`	no	no
`gdb`	`root:sys`	`rwxr-xr-x`	yes	yes
`home`	`bin:bin`	`rwxr-xr-x`	yes	yes
`local`	`daprog:daprog`	`rwxr-xr-x`	yes	yes
`lost+found`	`root:root`	`rwxr-xr-x`	no	no
`pgplot`	`root:users`	`rwxr-xr-x`	yes	yes
`pointing`	`daprog:observer`	`rwxrwxr-x`	no	no

Two of these directories are system directories and should not be touched. They are `TT_DB` and `lost+found`.

`/aux2/Drp`

This directory contains the source code and scripts needed for the `drp` data reductoin software. The start script itself is located in `/aux2/Drp/SCRIPTS/drp`. All files are owned by `root:root` and the size of the directory is about 6.5 MB.

`/aux2/TT_DB`

This is a system directory containing data for the authentication database. It contains 10 files which all are in binary format and are owned by `root:root` with permission `rw-r--r--`. The size of the directory is about 20 kB.

`/aux2/bin`

This directory contains a set of programs like `xspec` and `xv`. Some of them are not used since they are actually installed in other places of the system. All programs are owned by `root:sys` and are only writable for the owner. The size of the directory is about 9 MB.

`/aux2/catalog`

This directory contains the LOVAS spectral line database. All the files are ASCII files owned by `root:sys` with permission `rw-r--r--`. The size of the directory is about 3.6 MB.

`/aux2/dvips`

This directory contains the distribution of the `dvips` program with source code and other files. The program itself is not available in this directory and it is unclear why `dvips` should be on VELA at all. All the files are owned by `root:sys` with permission `rw-r--r--`. The size of the directory is about 2.4 MB.

`/aux2/gag`

This directory contains the `GILDAS` data reduction software distribution. Programs and scripts are located in the directory `/aux2/gag/hpux/bin` including, for example, `class`. All files are owned by `root:sys` and are only writable for the owner. The size of the directory is about 35 MB.

`/aux2/gcc`

This directory only contains two `tar` archives containing `gcc` software. They are owned by `root:sys` with permission `rw-r--r--`. The size of the directory is about 9.1 MB.

`/aux2/gdb`

This directory contains the distribution for the GNU development tools. All files are owned by `root:sys` and are only writable for the owner. The size of the directory is about 24.4 MB.

`/aux2/home`

This directory is used for the home directories of all Pegasus accounts. It only contains one subdirectory called `daprog` which is the home director for `daprog`. All the other Pegasus account then have their home directories located in `/aux2/home/daprog`. The `daprog` home directory is owned by `daprog:observer` and has permission `rwxr-xr-x`, while all other home directories are owned by their account and have permission `rwxrwxr-x`. This is necessary in order for `daprog` to be able to modify files for the different Pegasus users. The size of the `daprog` directory, *excluding* the home directories for the other Pegasus accounts, is about 2.4 MB.

The `daprog` home directory contains a lot of old crap which probably can be deleted without anyone missing it. However, there are a few important things worth mentioning. The first is the file `cronfile` which contains the `cron` job run by `daprog` once a minute to retrieve weather data from the Rosasix weather station (for `cron` jobs, see the section "*Cron jobs*" in the **SEST system administration handbook**). The second is the subdirectory `System` which contains a set of scripts that is needed for `daprog`, such as `create_argo_task` and `pcp`.

For a description of the directory structure of the Pegasus observing accounts residing below `/aux2/home/daprog`, see the section called "*The layout of an observing account*".

`/aux2/local`

This is the directory that holds the entire Pegasus system. It contains two subdirectories: `cfht` and `man`, both owned by `daprog:daprog` and having permission `rwxr-xr-x`. The first of them actually contains the Pegasus system and has a total size of 450–500 MB. The name "`cfht`" comes from the origin of the Pegasus system at the *Canada–France–Hawaii Telescope*. The structure of this directory is described further in the section "*Directory structure of* `/aux2/local/cfht`".

The `man` directory is quite small, about 270 kB, and contains three subdirectories: `man1`, `man3` and `man4`, all with the same ownership and permission as their parent directory. This is the directory structure of the Pegasus `man`–pages. All three subdirectories contains files in a format useable by the `man` command. All files are owned by `daprog:daprog` and have permission `rw-r--r--`. The `man1` directory contains `man`–pages on Pegasus programs; most of them being obsolete since they describe programs that existed at CFHT. The `man3` directory contains `man`–pages for a large set of Pegasus library functions (mainly `cfht_` and `cfp_` functions) and is thus useful for anyone working with Pegasus software. The `man4` directory only contains three `man`–pages but they are very useful:

- `cfp.4` which describes how to use the *configuration file parser*, i.e. the `cfp_`–library that handles `.par`–files including the syntax of variables in `.par`–files
- `ph.4` which describes the Pegasus help system including how to use it in `.par`–files
- `xform.4` which describes the syntax and usage of `.par`–files for the graphical menu windows

`/aux2/lost+found`

This is a system directory that is empty unless there is a problem with the disk, in which case orphaned files are put here.

`/aux2/pgplot`

This directory contains the `PGPLOT` distribution. All files are owned by `root:sys` and are only writable for the owner. The size of the directory is about 10.4 MB.

`/aux2/pointing`

This directory contains log files with data for all pointings and focussings that have been done. The data is stored in daily ASCII files which are named `pointing.`*YYMMDD* and `subpointing.`*YYMMDD*, respectively. The files are owned by different Pegasus observers, but all belong to the group `observer` and have permission `rw-rw-r--`. This directory grows constantly with approximately 1 MB per month. It should probably be transferred to the `/logs` partition.

`/aux2` on carina

The `/aux2` on `carina` is a mirror of the `/aux2`–disk on VELA updated on a bi-weekly schedule, with one exception: it also contains a directory called `Acroread3` owned by `gpersson:users` and having permission `rwxr-xr-x`. It contains the installation package for the Acrobat reader as well as the program itself (`/aux2/Acrobat3/bin/acroread`).

1.2.2 `/archive` (`/vela_archive`)

The `/archive` (as well as the `/vela_archive` on PUPPIS) partition is owned by `archive:users` and has permission `rwxr-xr-x`. It contains the on-line observing archive. Each observed project is stored as a `gzip`-ed `tar`-file in a subdirectory named after the month and year when the observation was archived (using the `archive` command). The directory names are of the form

MmmYY and have the same ownership and permissions as the parent directory. The individual files have names of the form *projectDDMmmYY*`.tar.gz`. They are also owned by `archive:users` but have the permission `rw-r-----`. This means that the archive can be accessed from staff accounts since they also belong to the group `users`, while observers' can't read the files (and steal other observers' data) since they belong to the group `observer`.

Except for the monthly archive directories there are also two other subdirectories in `/archive`: the mandatory `lost+found` owned by `root:root` and the normally empty subdirectory `incoming`. The latter one is owned by `archive:observer` and has permission `rwxrwxr-x`, and is thus the only part of the `/archive` partition where observers have both write and read access. This directory is used by the `archive` script to temporary store a new archive (in a subdirectory with the proper *MmmYY* name). The archiving daemon, which is run once every hour, will then change the ownership of the new archive file and move it to its proper destination, creating a new monthly directory if necessary.

There is normally no need to modify anything on `/archive` except when old archives are removed after having been transferred to CDs once every 6 months. Otherwise, directories and files are handled fully automatically by the `archive` script and the archiving daemon.

1.2.3 /logs

The `/logs` partition contains 5 log directories and the mandatory system directory `lost+found`. The top directory `/logs` is owned by `root:root` and has permission `rwxr-xr-x` which restricts the usage of the top directory to `root`. However, two of the subdirectories (the ones that are constantly used by Pegasus) have permissions that allow anyone belonging to the group `observer` to write to them.

Since log files have a tendency to grow infinitely in size, there is a need to do some maintenance on the `/logs` directory from time to time. This can, however, be done rather infrequently, say every 6 months. What has to be done is described further in the section on "...".

The log directories are:

Name	**Owner**	**Permission**	**Subdirectories**
`aos-logs`	`daprog:observer`	`rwxrwxr-x`	no
`communication-logs`	`daprog:observer`	`rwxrwxr-x`	no
`database-logs`	`root:sys`	`rwxr-xr-x`	no
`lost+found`	`root:root`	`rwxr-xr-x`	no
`odometer-logs`	`daprog:observer`	`rwxr-xr-x`	no
`pegasus-logs`	`root:sys`	`rwxr-xr-x`	no

Note that instruction file for `update_mirror` (`/var/adm/cpiofiles/update_mirror.disk`) as well as the statistics file (`/var/adm/cpiofiles/update_mirror.dirs`) may have to be changed, if directories or contents on `/logs` are changed!

/logs/aos-logs

The `aos-logs` directory contains log files for the channel drift of the three AOSs. There are three active log files, one for each AOS: `hrs.log`, `lr1.log` and `lr2.log`. These files must be owned by `daprog:observer` and have permission `rw-rw-r--`. They grow with 10–100 kB per month. Several old log files are also stored in this directory with the date of the last entries appended to the filename. These files have permission `rw-r--r--`.

/logs/communication-logs

The `communication-logs` directory contains two log files: `iccabin_server.log` and `pipe_flush.log` which contain the logs from the two programs with the same names, both belonging to the Pegasus task `socket_server`. These files grows slowly with time. Both must be owned by `daprog:observer` and have the permission `rw-rw-r--`. They contain error messages resulting from communication problems between VELA and ICCABIN.

/logs/database-logs

The `database-logs` directory contains one big called `db.out` that is owned by `database:users`. This is an old file from 1999 of unknown origin and content that were sitting on `/tmp` and was moved to this directory during a clean-up of `/tmp`.

/logs/odometer-logs

The `odometer-logs` directory contains compressed old odometer log files that are moved to this directory from `/aux2/local/cfht/dev/conf/odometer.log` by the system administrator when the odometer is reset (see the section "...").

/logs/pegasus-logs

The `pegasus-logs` directory is only used for short term storage of copies of the Pegasus log. A `cron`-job run daily by `root` checks the age of the file `/tmp/pegasus.log` and if it is older than 2 days, the Pegasus log (`/aux2/local/cfht/dev/conf/syslog.@`) is copied to this directory with a date and time stamp added to the name. The system administretor regularly processes these files and puts them into the permanent Pegasus log archive on `/logs/pegasus-logs` on CARINA (see the section on "`/logs` *on* CARINA" further down).

`/logs` on carina

There is also a partition called `/logs` on CARINA. It is owned by `root:root` and has permission `rwxr-xr-x`. Besides the mandatory `lost+found` directory it only contains two directories: `pegasus-logs` and `work`. The latter one is a work directory that normally only contains one file: `last_retrieved_log`. This file is used by `extract_pegasus_logs` to keep track of the date of the last incomplete Pegasus log stored on CARINA. It is a two-line ASCII file, with the first line containing the string "`# Pegasus log extractor date file`" and the second line contains the date in the format "*YYYY MM DD*".

Daily Pegasus log files are stored in `pegasus-logs` with names of the format `pegasus_log.`*YYMMDD*. These files contains logs in increasing chronological order, in contrast to the original log file which uses reverse order. The oldest logs are from the end of October 1999 and there are some gaps in coverage since then. Files that do not cover a complete day contains three lines of "`...`" where data is missing.

To retrieve Pegasus log files from VELA and put them on CARINA, hte program `extract_pegasus_logs` has to be run manually by `root` on CARINA. This program retrieves all log files from VELA, reformats them into daily log files, stores them in `/logs/pegasus-logs` on CARINA and then removes the processed files from `/logs/pegasus-logs` on VELA.

All directories and files on `/log` on CARINA are owned by `root:sys`. They are only writable for `root` but are world-readable.

1.2.4 `/cdscr`

There is no directory structure in the `/cdscr` partition since this is a scratch area, except for the mandatory system directory `lost+found` which is supposed to be empty. Any other files or directories found in `/cdscr` can be safely erased. The size of `/cdscr` is 1.5 GB.

Since this is a scratch area where anyone should have write permission, `/cdscr` is owned by `root:root` and has the permission `rwxrwxrwx`. This also means that anyone has the privilege to overwrite or erase any file regardless of ownership and permissions.

1.2.5 `/tmp`

The `/tmp` area is not a partition, but is just a directory on the root partition. It also contains a number of Pegasus related files. Some of them are rather critical to the system and may cause Pegasus to misbehave if they are altered or deleted. Care should thus be taken whenever files on the `/tmp` area are manipulated.

...

1.3 Directory structure of /aux2/local/cfht

The /aux2/local/cfht directory is the base directory for the entire Pegasus system including all the source code, libraries, programs and configuration files.

This base directory contains the following directories and files:

Name	Owner	Permission	Subdirectories One level	Subdirectories Nested
Doc	daprog:daprog	drwxr-xr-x	yes	yes
Runme	daprog:daprog	-rwsr-xr--	—	—
Runme.OLD	daprog:daprog	-rwSr-xr--	—	—
bin	daprog:daprog	drwxr-xr-x	no	no
conf	daprog:daprog	drwxr-xr-x	no	no
dev	daprog:daprog	drwxr-xr-x	yes	yes
ephlib	daprog:daprog	drwxr-xr-x	no	no
include	daprog:daprog	drwxr-xr-x	yes	no
lib	daprog:daprog	drwxr-xr-x	no	no
poincat	daprog:daprog	drwxr-xr-x	no	no
starcat	daprog:daprog	drwxr-xr-x	yes	no
tunefiles	daprog:observer	drwxrwxr-x	yes	no

The files Runme and Runme.OLD are compiled programs. Their function is unknown and their permission are suspicious with the suid–bit set.

1.3.1 /aux2/local/cfht/Doc

The Doc directory contains two Postscript files describing the Pegasus system as well as three subdirectories. The two Postscript files have been compressed with gzip and are called OwnersManual.ps.gz and UsersManual.ps.gz. These documents represents an early version (1991) of the original Pegasus system at the *Canada-France-Hawaii Telescope*.

The three subdirectories (libcfht, libui and ui) all contain a subdirectory called man. These man subdirectories contain a number of man–pages. Most of the files seem to be similar to the files stored in the /aux2/local/man/man*X* directories.

The total size of Doc is about 430 kB.

1.3.2 /aux2/local/cfht/bin

The bin directory is one of the five global Pegasus directories (see section "..."). This directory contains all compiled Pegasus programs as well as scripts and Tcl/Tk scripts.

All files in `bin` follow the ownership and permission rules that apply to global Pegasus directories with three exceptions: `newdata`, `roll` and `unroll` should all have the permission `rwsr-xr-x`. The total size of `bin` is about 60–100 MB.

1.3.3 `/aux2/local/cfht/conf`

The `conf` directory only contains one file and one named pipe related to the Pegasus log system (section "*The Pegasus log*"). The log is contained in a fixed size (10 485 766 bytes) ASCII file caleed `syslog.@` which should have the permission `rw-rw-r--`. This file is actually hard linked to the file `/aux2/local/cfht/dev/conf/syslog.@`. The named pipe is called `syslog.np` and has permission `prw-rw-rw-`. Both files must be owned by `daprog:observer` (and not `daprog:daprog` unless the permission of `syslog.@` is changed to `rw-rw-rw-`). The total size of `conf` is about 10 MB.

1.3.4 `/aux2/local/cfht/dev`

The `dev` directory contains subdirectories for all the Pegasus source code, all Pegasus tasks, all object files (`.o`–files) as well as one of the five global Pegasus directories. This is the subdirectory where all software development takes place and it will be discussed in further detail in the section "*Directory structure of* `/aux2/local/cfht/dev`". The total size of `dev` is about 380–420 MB.

1.3.5 `/aux2/local/cfht/ephlib`

The `ephlib` directory is one of the five global Pegasus directories (see section "..."). This directory contains the ephemeris databases needed by the `ephem` task to calculate the position of the planets and the moon. The ephemeris database is called `DEc405`. The files `iers.tab` and `leapsec.tab` are needed for moon calculations. The file `saveph.hp` is the old ephemeris database used by the now obsolete `ephemeris` task.

All files in `ephlib` follow the ownership and permission rules that apply to global Pegasus directories. The total size of `ephlib` is about 4.6 MB.

1.3.6 `/aux2/local/cfht/include`

The `include` directory contains one subdirectory called `cfht` which is one of the five global Pegasus directories (see section "..."). This subdirectory contains all globally installed C include files (`.h`–files).

All files in `include/cfht` follow the ownership and permission rules that apply to global Pegasus directories. The total size of `include` is about 500 kB.

1.3.7 /aux2/local/cfht/lib

The `lib` directory is one of the five global Pegasus directories (see section "..."). This directory contains all Pegasus library files (`.a`–files).

All files in `lib` follow the ownership and permission rules that apply to global Pegasus directories. The total size of `lib` is about 850–950 kB.

1.3.8 /aux2/local/cfht/poincat

This is an empty directory that could probably be removed.

1.3.9 /aux2/local/cfht/starcat

The `starcat` directory contains a set of source list. Three of them (`bolometer.list`, `cal.list` and `point.list`) are the catalog files that are available in the Pegasus `Source` menu. There is also a subdirectory called `oso` containing source lists from Onsala. All files are owned by `daprog:daprog` and have permission `rw-r--r--`. The total size of `starcat` is about 230 kB.

1.3.10 /aux2/local/cfht/tunefiles

The `tunefiles` directory contains 5 subdirectories, one for each of the five spectral line receivers: `rx100`, `rx115`, `rx150`, `rx230` and `rx345`. These subdirectories are owned by `daprog:observer` and have permission `rwxrwxr-x`, in order for any observer to save tuning files into them. Tune files are ASCII files owned by the observer who creates them, belonging to the group `observer` and having permission `rw-rw-r--`. They can be written and read by the `tune` task. Each is about 470 bytes large and contains a list of tuning parameters. The total size of `tunefiles` is about 20 kB.

This directory should probably be moved to `/logs`, since it is a directory where observers can add their own files. From the small number of tuning files stored here, the conclusion seem to be that this is not a commonly used feature.

1.4 Directory structure of /aux2/local/cfht/dev

The `/aux2/local/cfht/dev` directory is the base directory for all the source code as well as the global directory for configuration files.

This base directory contains the following directories, links and files:

Name	Owner	Permission	Subdirectories	
			One level	**Nested**
`HereIAm`	`daprog:daprog`	`-rwxr-xr-x`	—	—
`Makefile`	`daprog:daprog`	`-rwxr-xr-x`	—	—
`README`	`daprog:daprog`	`-rw-r--r--`	—	—
`Synopsis`	`daprog:daprog`	`-rw-r--r--`	—	—
`Synopsis.cfht`	`daprog:daprog`	`-rw-r--r--`	—	—
`argo`	`daprog:daprog`	`drwxr-xr-x`	yes	yes
`bin`	`daprog:daprog`	`drwxr-xr-x`	no	no
`cfht.COPYING`	`daprog:daprog`	`-rw-r--r--`	—	—
`conf`	`daprog:daprog`	`drwxr-xr-x`	no	no
`epeus`	`daprog:daprog`	`drwxr-xr-x`	yes	yes
`jason`	`daprog:daprog`	`drwxr-xr-x`	yes	yes
`medusa`	`daprog:daprog`	`drwxr-xr-x`	yes	yes
`obs`	`daprog:daprog`	`drwxr-xr-x`	yes	yes
`pegasus`	`daprog:daprog`	`drwxr-xr-x`	yes	yes
`src`	`daprog:daprog`	`lrwxr-xr-x`	—	—

1.4.1 `/aux2/local/cfht/dev/argo`

1.4.2 `/aux2/local/cfht/dev/bin`

1.4.3 `/aux2/local/cfht/dev/conf`

1.4.4 `/aux2/local/cfht/dev/epeus`

1.4.5 `/aux2/local/cfht/dev/jason`

1.4.6 `/aux2/local/cfht/dev/medusa`

1.4.7 `/aux2/local/cfht/dev/obs`

1.4.8 `/aux2/local/cfht/dev/pegasus`

Chapter 2

Pegasus tasks

2.1 Creating a new Pegasus task

If new functionality is to be added to the Pegasus system, it may be appropriate to create a new task for it. If new software has a natural place in an already existing task, it should probably be added to that task. There are also some tasks that contain software that are not big enough to motivate a dedicated task. These are the `@/misc` and the `@/utilities` task. The reason for having two such tasks is that the `@/utilities` task is a new task that was added by me which only contains software that really is in use, while the `@/misc` task contains a lot of old crap. The long-term goal is to clean up in `@/misc` by moving the relevant software to `@/utilities` and then remove `@/misc`. The policy should be to have many small and well-defined tasks rather than few and over-filled ones.

2.1.1 Creating the directory

This is most easily done by using the script `create_argo_task` which is available on the `daprog` accounts on VELA and CARINA. The script resides in `daprog/System`.

> `create_argo_task` *new-task-name*

This script will create a directory for the task in `/aux2/local/cfht/dev/argo` and install all necessary subdirectories and links. The script will also launch an `emacs` window with the script `@/Compile_ALL` where the new task should be manually included in the list of tasks.

Note that all tasks must have a name beginning with a lowercase letter.

2.1.2 Adding files to the directory

Note that the only files allowed in the main directory of a task are the program source code, scripts, the makefile and the `Usage` file describing the dependencies within the task. Any other files, such as documents, source code not in use and test programs should be put into properly named subdirectories. Neither object files nor compiled programs should reside in the main directory. They will be created in the subdirectory `bin` (which actually is a link to another directory structure). In other words: only files necessary for the compilation and installation of the task should be located in the main directory. Likewise, the only files allowed in the subdirectory `conf` are `.par` files, `.par_usage` dependency files, bitmap files or other files that should be installed globally.

For example, the `@/ephem` task contains a `tar` file of the original package received from Onsala. This `tar` file has been put into a subdirectory called `Archive`. The `@/ephem` task also contains software and data files needed to generate the ephemeris database. Since this software was only needed once, when the `@/ephem` task was first installed, all these files reside in another subdirectory called `Prepare` which has a standalone `Makefile` that can be used in case this database needs to be rebuilt. This `Makefile` then puts the database in the `@/ephem/conf` directory for installation by the main task `Makefile`.

2.1.3 Directory structure of each Pegasus task

The main directory for each task will be a directory with the name of the task located in the Argo base directory (`/aux2/local/cfht/dev/argo`). It must contain the following directories, links and files:

- `Makefile` (file): a makefile that should follow the standard format as much as possible. It must support the commands "`make all`" which should compile all the source code, and "`make install`" which should install all compiled programs, scripts, include files and configuration files (residing in the `conf` subdirectory) in the global Pegasus directories.
- `Usage` (file): a file that in a standardized way describes all components of a task and their dependencies. This file has to be created and updated manually each time anything is changed.
- `bin` (symbolic link): a symbolic link pointing to "`../../obs/`*task*`/bin`".
- `cfht` (symbolic link): a symbolic link pointing to "`.`".
- `conf` (directory):
- `conf/previous` (directory): a directory which contains copies of all old versions of parameter files, bitmap files and other files in the `conf` directory. Files are automatically copied here when the `pcp` script is used.
- `include` (symbolic link): a symbolic link pointing to "`.`".

- `man` (directory): a directory for `man` pages written in a format that can be read by the UNIX `man` command. As most people find `man` pages extremely boring both to write and to read, this directory is usually left empty.

- `previous` (directory): a directory which contains copies of all old versions of makefiles, source code, include files and scripts. Files are automatically copied here when the `pcp` script is used.

Object files ("`.o`–files") and compiled executables are automatically put in the directory pointed to by `bin` in the list above. This directory is also called `bin` and resides in `/aux2/local/cfht/dev/obs/`*task*. The immediate parent directory to `bin` has the same name as the task and does not contain anything else than the `bin` directory.

When the `create_argo_task` script is used to create a new task all the directories and links discussed in this section are automatically created ... and a template Makefile is ...

Safety copies of files

Whenever a file is modified, a safety copy of it should first be transferred to the appropriate `previous` directory. This should be done with `pcp` script:

> `pcp` *file-name*

This script preserves the last modification date of the file being copied and adds it as a suffix to the file name in the *YYMMDD* format. The `pcp` script will automatically create a `previous` directory, if there is none. Files should NOT be added to the `previous` directories in any other way except using the `pcp` script.

The use of the `previous` directories (and the `pcp` script) started in late 1999. Versions of files that are older than that have not been put in the `previous` directories. Instead, a number of tasks contain subdirectories called `backup` and/or `conf/backup` which contain these files (without proper date tags).

Main directories in a task

There are thus 3 important directories for each task: its main directory (`@/`*task*), its `conf` subdirectory and its `bin` subdirectory.

- `@/`*task*: This directory should only contain source code and include files used in programs that are installed and used globally, script files used globally, a makefile and a standardized documentation file called `Usage`. Any other files (documentation, obsolete source code, test programs not meant for global usage and so on) should all be put in appropriately named subdirectories. Programs and scripts should preferably not begin with a capital letter, while non-standard subdirectories preferably should.

- `@/`*task*`/conf`: This directory should only contain parameter files (`.par`) and their documentation files (`.par_usage`), bitmap files (`.bm` or `.xbm`), configuration files (`.cfg`), definition files (`.def`), resource database files (`.rdb` or `.xrdb`) and message files (`.msg`). Any other file should be put into a properly named subdirectory. The same should also be done for files of any of the allowed types that are not in current global use.

- `@/`*task*`/bin`: (actually a link to `@/../obs/`*task*`/bin`) This directory only contains files that are created when "`make all`" is run. No other files should be put there. If you want to recompile a program, the proper thing to do is to delete the files in this directory before running "`make all`".

2.2 Compiling and installing a Pegasus task

There is a `Makefile` in the main directory of each Pegasus task that has to be used when compiling and installing a Pegasus task. Other methods of compiling and installing programs and files are not allowed. Note that the usage of this makefile requires the use of `csh`! If you try to use `make` when using `tcsh`, you will only get the error message: "`Make: line 106: syntax error. Stop.`".

If modifications have only been made to files that do not need to be compiled before installation, like scripts and par–files, there is no need to (re-)compile the task. However, for any change to be made globally, the modified files will still have to be properly installed as described below.

2.2.1 Compiling a task

The procedure to compile a Pegasus task is to move to the main directory of the task (`@/`*task*) and then issue the command:

> `make all`

This will compile all programs belonging to the task, provided that they have been modified since the last time they were compiled. To only recompile a certain program, use:

> `make` *program-name*

To force a program or program part to be recompiled even if `make` does not want to recompile it, you can delete the proper files from the `bin` subdirectory. To force a recompilation of everything, you can use the command

> `make clean`

which not only will delete all executables and object files in the `bin` subdirectory, but will also erase the directory itself and the link to it. The directory and the link will be automatically restored by the next "`make all`" command.

Whenever, you change a file, you should check the `Usage` file, to see if this file is also used by other tasks. In that case you will have to recompile and reinstall these tasks as well afterwards. If there are several other tasks to be recompiled (which certainly is the case if you change the `libsest` task, for example), or if you feel uncertain whether the `Usage` file really is up-to-date, you can recompile all tasks. This should not take too much time, since nothing will be done to those tasks that already are up-to-date. The instructions for recompiling all Pegasus tasks can be found in the section "Recompiling and reinstalling all Pegasus tasks".

2.2.2 Installing a task

To be available globally to any Pegasus user, programs and files have to be installed in some special global directories. To install modified programs, scripts, `.par`–files and so on into these directories, use the command:

> `make install`

Only programs and files that have been changed since the last compilation and installation will be affected.

If you are installing a modified `.par`–file, you should also delete all the local copies of this file that resides on all the observing accounts, since the global file is not used when there is a local copy of the file available. New local copies are created automatically from the global one as soon as they are needed. The command to use is:

> `rms .,`*filename*`.par`

This script will go through all observing directories, look for the specified file and ask if it is OK to delete if it exists. Remember that the name of a local version of a `.par`–file always begins with "`.,`"!

There are 5 global directories, which are used for different types of files. They are:

- `/aux2/local/cfht/bin` is the directory where all programs as well as all shell scripts and Tcl/Tk scripts are installed in.
- `/aux2/local/cfht/lib`: is the directory used for libraries.
- `/aux2/local/cfht/ephlib`: is a special library with files needed for the planetary ephemeris (only used by the `ephem` task).
- `/aux2/local/cfht/include/cfht`: is the directory where all include files ("`.h`") are put.

- `/aux2/local/cfht/dev/conf`: is the directory containing all configuration files, i.e. all files that comes from the `conf` directory of the task ("`.par`", "`.bm`", "`.xbm`", "`.cfg`", "`.def`", "`.rdb`", "`.xrdb`" and "`.msg`" — but not "`.par_usage`") — this directory also contains files for the odometer and the Pegasus log.

The installation of a file in one of these global directories is done in the following way by "`make install`":

- The file is copied to the global directory and installed with a dash and a date appended to its name (*file*-*YYMMDD*).
- A hard link to this file is created in the same directory with the same name but without any suffix.

The idea behind this scheme is that this will keep older versions of programs and files available in the global directories, in case it turns out that something is wrong with the modified file and you want to go back to the earlier file. How to do this is described in the section called "Restoring an old Pegasus file".

This scheme makes it necessary to sometimes clean up the global directories by removing all old and no longer used files. This can easily be done with the procedure described in section "...".

Note that it is NOT allowed to install programs and files in any other way than by using "`make install`". To directly copy or modify a file in one of the global directories mentioned above is strictly forbidden!

2.2.3 Restoring an old Pegasus file

However, the use of this procedure should be avoided, since the file in the global directory then will not represent the state in the main directory of the task. Strange and unexpected results may then appear the next time someone modifies something in that task.

The recommended way is instead to go to the main directory of the task and restore the file from the safety copies kept in the local `previous` directory. Use the script `...` by typing:

> `...` *filename*

and select the proper file to restore from the list.

After you have restored the file, you will have to recompile the task, if it was a program source code file, with "`make all`" and reinstall it again with "`make install`". If you restored a `.par`–file, you should also remove all local versions of the file with "`rms .,`*filename*`.par`".

2.3 Recompiling and reinstalling all Pegasus tasks

There is a simple way of recompiling and reinstalling all Pegasus tasks at once. This is especially useful whenever a library has been modified, since this requires the recompilation of all tasks that use that library. Any changes to the `libsest` library will, for example, require most tasks to be recompiled. Another situation when a total recompilation may be useful is when you are uncertain about whether other tasks are depending on a file you have modified. Since nothing will be done to all tasks that already are up-to-date, it does not take more time to actually ask for a total recompilation.

To recompile and reinstall every task that is out-of-date, go to the Argo base directory (`/aux2/local/cfht/dev/argo`) and use the command `Compile_ALL`.

2.4 Adding software to an existing Pegasus task

2.4.1 Adding a new program

2.4.2 Adding a new script

2.4.3 Adding a new parameter file

2.5 Modifying a Pegasus task

2.5.1 Modifying a program or a script

2.5.2 Modifying a parameter file

Chapter 3

Modifying the Pegasus menu or other configuration files

Chapter 4

Observing with Pegasus

4.1 The layout of an observing account

4.2 Login sequence

4.3 Archiving and cleaning an account

Whenever an observing project has been finished, the data should be archived and the account cleaned, before the account is being used for the next project. The simplest way to do this is to use the menu you get when press down the left or the right mouse button in the background. The submenu "`For staff only`" contains "`Archive`" and "`Clean Account`". It is also possible to open a shell window and use the scripts `archive` and `clean_account`, respectively. Both archiving and cleaning can be done from any Pegasus account, i.e. it is not necessary to log in to the specific account you want to archive or clean.

The corresponding guest account on PUPPIS should also be cleaned when a project has been finished. This is done by logging in on PUPPIS as `sest`, move to the subdirectory `guest_setup` and run the script:

> `clean` *account-name* `'`*account-password*`'`

4.3.1 Archiving

The `archive` command (belonging to the `utilities` task) collects all FITS files and stores them in a `tar` file which is compressed using `gzip`. This file is then appended to the monthly archive DAT–tape and put in the directory `/archive/incoming/`*MmmYY*, where the archiving daemon will find it and store it in the on-line archives on VELA and PUPPIS.

The `archive` script checks the home directory of the specified user for the existence of two files: `archive.id` and `project.id`. Both these two files are created by the `project` script (belonging to the `project` task) the first time a user logs in to an unused Pegasus account. If `project.id` does not exists, archive will assume that the account is empty and will not try to archive it. If the project.id file for some reason is missing, you can recreate it by running "project".

The file archive.id is removed by archive after an account has been archived. This is done to prevent archive from being run twice on the same account, since archive will not run if archive.id is missing. If the archive.id file for some reason is missing, you can recreate it by typing "touch archive.id".

4.3.2 Cleaning on vela

The clean_account script (belonging to the utilities task) automatically erases all files of certain types, for example, all numbered FITS files, any Postscript file and any user-defined map-file. It also erases all files residing in any of the default subdirectories: online, shadow and temp, as well as all .par–files. It then checks the existence of all necessary files and directories beginning with a character, for example all cur$x$.fits files, and install them if necessary by copying them from the directory @/Default_files. After that all remaining files and directories beginning with a character that do not belong to the default set-up are listed and the user is asked whether it is OK to delete them. Finally it checks that all necessary system files starting with a "." exists and have not been modified by comparing them to the default versions stored in @/System_files. This is done in a clever way since some of the system files are customised for each user. For these files, clean_account first makes a customised version from the file stored in @/System_files before actually doing the comparison with the file in the user's account. Any modified file is automatically replaced.

4.3.3 Cleaning on puppis

The clean script cleans a Pegasus account by removing everything from the account and then reinstalling the necessary system files and directories from the tar–file /home/sest/guest_setup/puppis_setup.tar. It also uses the files raw.bookmarks.html, raw.preferences, raw.userpattern and rhosts located in the same directory. The process is done with a combination of rsh and ftp commands to do the job. For this scheme to work, the .rhosts file on the account to be cleaned is set-up to allow the user sest to access the guest account without having to provide a password.

There is one problem related with clean: if an incorrect password is given to clean, the script usually manages to remove everything from the specified account, but it will fail when trying to reinstall the system files (producing a series of error messages in the process). It will then still be possible to login to the cleaned account but a lot of things will not work properly. This is simply cured by running `clean` again with the correct password.

Chapter 5

Special Pegasus features

5.1 The Pegasus log

Everything that is sent to the Pegasus feedback window together with a lot of information not sent to this window end up in the Pegasus log. The `Log` button in the Pegasus meny lists this log in time-reverse order for the particular user being logged in.

The log is stored in a file called `syslog.@` which can be found in two locations: in the general configuration directory (`/aux2/local/cfht/dev/conf`) and in another configuration directory that does not contain anything else than the log (`/aux2/local/cfht/conf`). These are actually hard links to the sama file, which should be owned by `daprog:observer` and have permission `rw-rw-r--`. In each of the two mentioned directories there is also a named pipe called `syslog.np` with the same ownership and the permission `rw-rw-rw-`.

The `syslog.@` has a fixed size of 10 485 766 bytes. The log is written to it in a rolling fashion so that the oldest parts of the log is overwritten by the latest entries. This is done in a time reverse order. The pointer to where the last data is written is thus working its way from the end of the file up to the top of it, before beginning again at the end of it. The file is in ASCII format although there are a few control characters just in the beginning of the file and where the current pointer is. It typically holds 3 to 4 days of log messages.

Whether it is necessary to have two entries to the Pegasus log and whether the two named pipes are really used need to be investigated.

5.2 The Pegasus log archive

An archive of old Pegasus logs have been established on CARINA. This can be used if older logs are needed for something, since the Pegasus log is overwritten after about 3–4 days. The logs are stored in daily files called `pegasus_log.`*YYMMDD* in the directory `/logs/pegasus-logs`. They are all owned by `root:sys` and have the permission `rw-r--r--`. The archiving of Pegasus logs started in the end of October 1999 and is rather complete since then, although there are a few gaps from the first 6 months. In contrast to the real Pegasus log, the data is stored in increaing time order in the daily logs. Any missing data in a file is indicated by three lines of "...". A typical daily log file has a size of a few MB.

A `cron` job owned by `root` checks the file `/tmp/pegasus.log` once a day. If it does not exist or is older than 2 days, a copy of the Pegasus log (`/aux2/local/cfht/conf/syslog.@`) is moved to the directory `/logs/pegasus-logs`. The name of the copy is changed to `pegasus.`*YYMMDD_hhmm* and the file `/tmp/pegasus.log` is updated.

Any files in `/logs/pegasus-logs` are transferred to the directory `/logs/work` on CARINA, whenever the system administrator runs the script `extract_pegasus_logs` as `root` on CARINA. This script then reads the file `/logs/work/last_retrieved_log` to find out the date of the last incomplete daily Pegasus log currently stored in `/logs/pegasus-logs` on CARINA. It then runs the program `extract_day` to extract daily Pegasus logs from the transferred log files starting with the date found in `/logs/work/last_retrieved_log` and continuing until there is no more data. The log entries are written to the daily logs in increasing time order. They are given names of the form `pegasus_log.`*YYMMDD* and put in `/logs/pegasus-logs` on CARINA. The original log files on the `/logs/pegasus-logs` directory on VELA are then automatically removed.

5.3 The odometer

The odometer is the facility that generates scan numbers for the FITS files. It consists of two files residing in the Pegasus global configuration diretory (`/aux2/local/cfht/dev/conf`): `odometer` and `odometer.log`. They must both be owned by `daprog:observer` and have permission `rw-rw-r--`. Both are ASCII files. The file `odometer` contains a single line with the scan number last used, while `odometer.log` contains a list of all scan numbers used so far, when they were used and by whom. The latter file rapidly grows by more than 1 MB per month.

The odometer should be reset once a year as described in the section "*Resetting the odometer*" and the old odometer log file is then compressed and moved to `/logs/telescope-logs`.

5.4 Archiving daemon

5.5 The alarm system

5.6 Wind autopark

5.7 Weather data

5.8 Time synchronisation

5.9 Communication with other computers

Chapter 6

Other topics

6.1 Files with special privileges

6.2 Observing accounts

6.3 Changing the default files installed when cleaning an account

6.4 The Pegasus login sequence

6.5 The Pegasus environment

6.6 Template files

6.7 Pegasus program philosophy

6.8 On-line documentation

6.9 Periodic maintenance tasks

The Pegasus system does not require very much maintenance besides what is described in the "*SEST system administration handbook*". However, there are a few tasks that should be performed now and then (usually once every 6 months or once a year) and they are described in this section. You should be logged in as `daprog` whenever you are performing any Pegasus maintenance.

6.9.1 Transferring the on-line archive to CDs

6.9.2 Cleaning up old Pegasus files

6.9.3 Resetting the odometer

The odometer providing the scan numbers to the FITS files should be reset once a year, preferrably when a new year starts. The procedure is as follows:

- Make sure that no observer is logged in to VELA.
- Go to the global configuration directory:
  > `cd /aux2/local/cfht/dev/conf`
- Edit the file `odometer` and reset the value to `10000`. Make sure that the file only contains this value and a new-line character.
- Move the odometer log file to `/logs/telescope-logs` for storage and add the current date as a suffix to the name:
  > `mv odometer.log /logs/telescope-logs/odometer.log-`*YYYYMMDD*
  > `gzip -v /logs/telescope-logs/odometer.log-`*YYYYMMDD*
- Create a new odometer log file:
  > `touch odometer.log`
- Make sure that both `odometer` and `odometer.log` have the owner `daprog:observer` and the permission `rw-rw-r--`:
  > `chown daprog:observer odometer odometer.log`
  > `chmod ug=rw,o=r odometer odometer.log`

6.9.4 Resetting Pegasus log files

Once or twice a year, the log files for the AOS drift and the communication to ICCABIN should be cleared. The procedure is the following:

- Make sure that no observer is logged in to VELA.
- Go to the communication log directory:
  > `cd /logs/communication-logs`
- Remove the old log files and create new, empty ones:
  > `rm iccabin_server.log pipe_flush.log`
  > `touch iccabin_server.log pipe_flush.log`
  > `chmod ug=rw,o=r iccabin_server.log pipe_flush.log`

- Move to the AOS log directory:
  > `cd /logs/aos-logs`

- Change the permission and rename the three log files `hrs.log`, `lr1.log` and `lr2.log` by appending the current date to their names:
  > `chmod u=rw,go=r hrs.log lr1.log lr2.log`
  > `mv hrs.log hrs.log_`*YYMMDD*
  > `mv lr1.log lr1.log_`*YYMMDD*
  > `mv lr2.log lr2.log_`*YYMMDD*

- Create three new log files and give them the proper permission:
  > `touch hrs.log lr1.log lr2.log`
  > `chmod ug=rw,o=r hrs.log lr1.log lr2.log`
  > `chown daprog:observer *`

6.10 Environment

The official shell to be used for Pegasus is `csh`. For this reason, all Pegasus accounts must have `csh` as their default shell. Most applications will also work well under `tcsh`, but as long as it has not been verified that the entire Pegasus system is `tcsh` compliant, `csh` must remain the default.

All Pegasus users belong to a special group called `observer` ...

6.11 Pegasus users

There are currently 11 different Pegasus accounts installed on VELA and CARINA, but only 9 of them are installed on PUPPIS (see table below).

ID	Username	Group	VELA	PUPPIS	CARINA	Password
101	`daprog`	`observer`, `daprog`	*	*	*	staff
102	`pegasus`	`observer`	*	*	*	staff
104	`blue`	`observer`	*	*	*	guest
105	`yellow`	`observer`	*	*	*	guest
106	`red`	`observer`	*	*	*	guest
107	`green`	`observer`	*	*	*	guest
108	`orange`	`observer`	*	*	*	guest
109	`violet`	`observer`	*	*	*	staff
110	`director`	`observer`	*	*	*	staff
115	`vlbi`	`observer`	*	-	*	staff
116	`puma`	`observer`	*	-	*	staff

The `daprog` account can belong either to the general Pegasus group `observer` or to its own group `daprog`. The idea behind this is probably that ... However, this has not been used. The default group for `daprog` when logging in on VELA and CARINA is `observer`, while it is `daprog` on PUPPIS. Parts of the Pegasus system belongs to the group `daprog` while other parts belong to `observer`.

Username	**Usage**
`daprog`	development account used for all software modifications — owns all Pegasus software — the only account that does not start Pegasus upon login
`pegasus`	staff observing account intended for live testing of software — has not been used
`blue`	guest observing account
`yellow`	guest observing account
`red`	guest observing account
`green`	guest observing account
`orange`	spare guest observing account
`violet`	special staff observing account
`director`	main staff observing account
`vlbi`	account intended for VLBI use — has not been used
`puma`	safe software testing account — a lot of Pegasus programs fake their action when run from `puma` to provide a safe test environment, however, not all programs are up to this standard, since this feature has not been used

6.12 The `daprog` environment

6.12.1 The `$path` variable

Besides standard ...

6.12.2 Aliases

There are a number of aliases defined in `daprog`'s `.cshrc` file, most of them are aliases for `cd` commands to various parts of the Pegasus system.