AEDIT version 4.1

A program for editing, manipulating, visualizing and summarizing
MkIII/IV A-file format data

by
Colin Lonsdale
Haystack Observatory

September 1994

New in 4.1

Previous to this release of aedit, there were three types of A-file lines, corresponding to the three types of binary data files, root (type 0), corel (type 1) and fringe (type 2). I have now added two new types of A-file lines. They are type 3 (closure triangle) and type 4 (closure quad) lines, and they can be formed only by combining multiple type 2 (fringe) lines. In this release of aedit I have implemented full support for type 3 lines only. Type 4 support will be added later at lower priority. Closure triangle data can now be read, written, filtered, plotted, and interactively edited in much the same way as type 2 data. You can get closure triangle data into aedit two ways. First, you can read it in from a disk file using the standard read command. Alternatively, you can read in some type 2, baseline data, and use the new close command to compute closure triangle data from the type 2 data. Either way, the closure data are fully plottable and manipulable. It is important to note that aedit is (and should remain) the only HOPS program capable of generating closure data. All closure triangle data in HOPS must ultimately originate from the close command in aedit operating on type 2 data, though it can be read, written and manipulated by aedit and other programs.

Because the data types are fundamentally different, type-3 closure triangle data and type-2 baseline data are stored internally in separate arrays. They should also be stored on disk in separate files, so there is now a new twrite command (analogous to the existing write command), which writes out only closure triangle data. If you really want to keep your baseline and closure data in one file, you can concatenate them and aedit will swallow the resulting file. However, since other HOPS programs (such as average) will be expecting one or the other, and not both, it will probably be easier to keep them separate.

To accommodate closure triangle data, there are a couple of new filters you can set. First, you can specify the closure triangles you want with the triangles command (station order is irrelevant), and second, you can specify a range of bispectral SNRs, using bsnrmin and bsnrmax. The bispectral SNR calculated by the close command is the theoretical one based on the individual baseline SNRs. If the data come from a file instead, the bispectrum SNR may have been calculated differently (e.g. by program average). Certain filters do not apply to certain data types, and are simply ignored. For example, saying triangles ABC; edit inputs will have no effect on baseline (types 1 and 2) data. All of the (relevant) filters familiar to you for baseline data, such as stations, quality code and so on work for closure triangle data.

There are some new options to the edit command which permit you to control the consistency of related baseline and triangle datasets. The edit close triangles command removes triangle records which do not have all three legs of the triangle present in the baseline dataset. The edit close baselines command, conversely, removes all baseline records which do not participate in at least one valid triangle record. Just typing edit close does both operations, and ensures a fully consistent set of baselines and triangles.

The unflag command has been enhanced to accommodate the new types of flagging made possible by these filtering and editing changes. Aedit makes no attempt to detect triangle redundancies (i.e. it will generate and/or accept N(N-1)(N-2)/6 triangles for N stations). While this may result in excessive amounts of closure triangle data for large experiments, it is necessary in order to preserve information in the low SNR case (where the triangles are not fully redundant), and besides simplifies what is already some quite convoluted logic for handling arbitrary input closure data files.

There have been a number of minor enhancements to the way various commands function. In all cases, these changes should cause no problems for existing aedit users.

New in 4.0

There have been many enhancements in "aedit" since the release 3.0 in March 1992, some of which are major. Below is a list of the most important enhancements. The detailed documentation of the individual commands covers the numerous other, more minor changes.

1. Full support is now available for all three types of data record, 0, 1 and 2. This applies to filtering, sorting, and other operations. Two new "edit" options are now available which remove data records

which lack parent or child data records (e.g. if a root record has no associated corel or fringe records in the database, it can be edited out with the command "edit parents"). The "family" sort option causes data records to be grouped by root family when writing the data to disk (normally, all type 2 records are written, then all type 1's, then all type 0's).

2. Aedit now has significant experiment summary capabilities. The command "ccread" causes a specified correlator control file to be read into memory, after which one may invoke the "psplot" command. This is a highly interactive experiment status browser and data editor, which uses color-coded rectangles to represent data records. You can tag or edit data based on quality code, scantime or baseline, or by clicking on individual data cells, and you can of course pop up fringe plots by clicking on any cell. The display is intended to visually assist diagnosis of experiment-wide problems, among other things.

3. In order to circumvent the problem of limited numbers of parameters in the standard A-file format, "aedit" now supports parameter extraction directly from the type-2 data files on disk. Depending on circumstances, you can now extract, plot, and write out any of a wide range of parameters. You can also filter the data on the basis of the values of extracted parameters. Phasecal information is now accessible only through the extracted parameters. The relevant commands are "parameter", "pwrite", "plist" and "prange".

4. Plotting within "aedit" has been improved significantly. It is now possible to plot any quantity, including extracted parameters if present, against any other quantity (provided it makes rudimentary sense .. you cannot plot closure rate versus baseline SNR, for example). Specification of the output device has been simplified for a subset of devices. All plots remain fully active, in the sense that on appropriate devices you can edit data with the interactive cursor, or pop up fringe plots on Xwindow devices. New commands of relevance include "xscale", "yscale", "reference" and "remote".

5. Along with all other MkIV programs, "aedit" supports the new version 2 A-file format. This format features a somewhat more useful mix of quantities, with additional precision in many cases, and is to be preferred where possible. "Aedit" can write the data out in any A-file format (see the command "outversion").

GENERAL DESCRIPTION OF AEDIT

The program "aedit" is a general purpose A-file manipulation program. The information present in one or more A-files may be plotted, filtered, sorted and edited in a variety of ways, before being written out in the form of a new A-file. The user interface to the program is presently implemented only for ASCII terminals, but many functions of the program interact with the user via a graphical interface.

Commands are given to aedit by keyboard, and a full minimum match capability is supported for all aedit names. Multiple commands are allowed on one input line, the only requirement being that commands are separated by the semicolon ";" character. Commands typically consist of the command name followed by 0 or more arguments. The arguments are separated from the command name and each other by either spaces or commas. Aedit can handle long lines, but it is of course bad practice to wrap lines on terminals in general. Upwards of about 250 characters may start to cause problems even for aedit. In general, aedit is not case sensitive. Case sensitivity is needed for UNIX filenames and for certain quantities from A-files (station codes, frequency codes, source names).

Aedit uses the concept of inputs. That is, you set up certain variables in the program that determine how the "action" commands will behave. Most of the commands that aedit understands are of the input-setting variety. Many are quite particular about syntax, and will complain if the user types nonsense (e.g. timerange). At any time, the current state of the input parameters can be listed on the screen with the command "inputs".

When aedit reads data from an A-file, it parses the ascii information and stores it as binary data in memory. This allows very rapid manipulation of the data once read in, with seemingly complex tasks appearing to be instantaneous. There is a flag field associated with each A-file line in memory, and these flags are manipulated by the edit and unflag commands. A full description of the data currently in memory can be obtained with the "summary" command. This is essential when deciding on plotting and editing options. The "write" command ignores flagged data, permitting aedit to be used as a simple and efficient filtering program.

A command "run" is available, which provides a flexible and general command file capability. Nesting of command files to a depth of 10 is allowed. The "run" command executes in batch mode, and cursor operations are therefore disabled.

Aedit features a shell escape. By starting an input line with the character "!", you can access standard UNIX commands outside of aedit. You can escape to a complete new shell by typing "!csh" or "!sh", and when you have finished, return to aedit where you left off by typing cntl-D. This feature is useful for spooling plot files to a printer, preparing run files, running "alist" to prepare new data for aedit, and any other tasks that you wish to perform without terminating the aedit session.

Plotting is implemented by using the PGPLOT package from CalTech. The output device may be specified with the "device" command, or you may leave "device" at the default value ("?"), which will cause PGPLOT to query you for a device at the time of plotting. Your response will then be automatically entered into the "device" input. A list of available device types can be obtained by responding with a query. For more information of devices, see "help device"

Aedit comes with full on-line help. In general, the syntax is "help command", but just "help" will work. The command line for aedit is "aedit [-x] [-r filename] [-f filename]", where "-x","-r", and "-f" are optional. The "-x" option means start up the xwindow interface (not yet supported). The "-r" option means execute the specified run file on startup, and must be immediately followed by the name of a file containing valid aedit commands. The "-f" option means "read this(ese) data file(s) on startup", and must be immediately followed by a standard, wildcardable UNIX filename specifier or specifiers. In this way, you can read many files at once into aedit without going through a laborious one-at-a-time "read" cycle within the program. If specified, the "-f" flag must be the last flag.

Below is a list of all current aedit commands:

```
Action commands:
----------------
batch        clear       close       edit         exit
fplot        help        inputs      nobatch      parameter
plist        plot        pwrite      read         run
setyear      sort        summary     twrite       unflag
unsort       write       zoom


Plot control commands:
----------------------
grid         xscale      yscale      axis         mode
reference    remote


Data selection commands:
------------------------
baselines    bsnrmin     bsnrmax     experiment   fraction
frequencies  length      nfreq       prange       procrange
qcodes       snrmax      snrmin      sources      stations
```

```
timerange    triangles   type


Experiment overview commands/parameters
---------------------------------------
schedread    psplot      psfile


IO control commands:
--------------------
device       outversion
```

For further information, see the individual help files for the above commands.

COMMAND DESCRIPTIONS

COMMAND **help**

TYPE **Action**

SYNTAX **"help 'subject'"**

DESCRIPTION

Writes the help file pertaining to "subject" on the terminal, under pagination control. "Subject" is presently any command name, plus "general".

Action commands:

```
Name      argument(s)      Description
----      -----------      -----------
clear     data             Erase all data from memory
          close Erase all  triangle data from memory
          inputs           Reset input settings to default
          plot             Clear screen or eject page
          all              All three clear functions
close                      Generate triangle data from type-2 data
edit      cursor           Zap points on screen with cursor
          inputs           Remove points that don't fit inputs
          duplicates       Remove duplicate points with various priorities
          parents          Remove childless parent records
          children         Remove orphan child records
          close triangles  Remove triangle record without baselines
          close baselines  Remove baseline record without triangles
          close            Perform both of the above commands
exit                       End aedit session
fplot                      Pop up Xwindow fringe plots
inputs    plot/filter      Print current input settings on screen
parameter number(s)        Extract parameters from type-2 files on disk
plist                      List extracted parameters in memory
plot                       Plot current data according to inputs
pwrite    filename         Write extracted parameters to filename
read      filename         Read in data from filename
run       filename         Execute commands in filename
setyear   number           Manually reset year of scan throughout data
sort      key              Sort data according to various keys
summary                    Display a summary of all unflagged data
twrite    filename         Write (edited, sorted) type 3 data to filename
unflag    string           Removes flags applied for various reasons
unsort                     Restore original sort order (as read in)
```

```
write   filename        Write (edited, sorted) type 0,1,2 data to filename
zoom                    Display details of cursor-selected points
```

Plot control parameters:

```
Name    argument(s)     Description
----    -----------     -----------
axis    string          Set variable to plot on Y axis
grid    a,b             Divide screen/page a times b subplots
xscale  min,max         Set X-axis scale
yscale  min,max         Set Y-axis scale
mode    split/nosplit   Do/don't do 1 plot per source
reference               Use reference antenna in baseline plots
remote                  Use remote antenna in baseline plots
```

Data selection parameters:

```
Name            argument(s)     Description
----            -----------     -----------
baselines       AB,BC,AC ....   Use only these type 2 baselines
triangles       ABC,DEF,ADE ..  Use only these type 3 triangles
experiment      expt #          Use only data from this experiment
frequencies     S,X,K, ....     Use only data at these frequencies
fraction        nn% ....        Use only scans with >nn% good data
length          number          Use only scans > number secs or blocks
nfreq           <>= nn          Use only scans with <>= nn frequencies
qcodes          5-9,D ...       Use only data with these quality codes
snrmax          number          Use only type 2 data with snr < number
snrmin          number          Use only type 2 data with snr > number
bsnrmax         number          Use only type 3 data with bsnr < number
bsnrmin         number          Use only type 3 data with bsnr > number
sources         name1,name2...  Use only data on these sources
stations        A,B,C,D ....    Use only data from these stations
timerange       yyddd-hhmmss, yyddd-hhmmss
                                Use only data in time range
prange          n, min, max     Use only data with parameter n in range
procrange       yyddd-hhmm, yyddd-hhmm
                                Use only data in procdate range
type            0 1 2 3         Use only data of these types
```

Experiment overview commands/parameters:

```
Name            argument(s)     Description
----            -----------     -----------
ccread          filename        Read specified CC file into memory
psplot                          Display active plot of experiment
psfile          filename        Write experiment summary to filename
```

I/O control parameters:

Name	argument(s)	Description
device	string	Plotting device for PGPLOT
outversion	number	Determines output A-file format version

Miscellaneous:

Name	argument(s)	Description
batch	none	Disables interactive confirmation querys
nobatch	none	Enables interactive confirmation querys

COMMAND **axis**

TYPE **Plot control**

SYNTAX **"axis y-axis <x-axis>"**

DESCRIPTION

This command specifies what the X and Y axes of the next "plot" command will be. The y-axis specifier is mandatory, the x-axis one optional. If the x-axis specifier is omitted, it is assumed to be "scantime". After the "axis" command, the "plot" command with no arguments causes the specified axes to be plotted on the current output device (see "help device"). If the "plot" command is issued with arguments, those arguments override the axis settings made by the "axis" command, and reset the plot inputs. The arguments to the "plot" command are identical to those described below. Axis specifiers are case-insensitive.

Valid axis specifiers, with notes, are listed below:

Minmatch string	Quantity	Notes
scantime (or time)	data time tag	default X-axis
pcal_phase(n)	Ref/remote pcal phase	Must extract first (see help param) n is integer array element index.
pcal_diff(n)	Ref/remote pcal phase diff	Must extract first Relative to channel 1
pcal_amp(n)	Ref/remote pcal amplitude	Must extract first Sign encodes pcal mode
error_rate	Ref/remote error rate	Must extract first
elevation	Elevation of telescope	If baseline plot,

		ref/remote controlled by "reference", "remote" commands
azimuth	Azimuth of telescope	Same as elevation
snr	SNR	
amplitude	Correlation amplitude	
phase	Residual scan phase	
sbdelay	Singleband delay	
mbdelay	Multiband delay	Ambiguities removed
drate	Delay rate	
cphase	Closure phase	Uses totals
crate	Closure rate	
csbdelay	Closure singleband delay	
cmbdelay	Closure multiband delay	Ambiguities removed
campl	Closure amplitude	NYI
u	U in megalambda	
v	V in megalambda	
uvdist	UV distance in megalambda	
param?	extracted parameter value	The '?' is an integer which specifies which parameter to use. To get a list, use "plist".

COMMAND **baselines**

TYPE **Data selection**

SYNTAX **"baselines AB BC CD AC"**

DESCRIPTION

Sets the baseline data selection parameter in the inputs. Only those baselines specified will pass the filter-applying operations of edit inputs, read, and plot. Typing "baselines" without arguments removes any restrictions on allowed baselines.

COMMAND **batch**

TYPE **Miscellaneous**

SYNTAX **"batch"**

DESCRIPTION

Disables confirmation mechanism, for running in batch mode. If you intend to plot data in batch mode, remember to set the plot device in your runfile, as the default PGPLOT query mechanism is disabled in this mode.

COMMAND **snrmax**

TYPE **Data selection**

SYNTAX **"snrmax 20"**

DESCRIPTION

Sets the maximum snr which will pass the filters applied in edit inputs, read, and plot. Typing "snrmax" without arguments removes any upper bound on snr.

COMMAND **snrmin**

TYPE **Data selection**

SYNTAX **"snrmin 20"**

DESCRIPTION

Sets the minimum snr which will pass the filters applied in edit inputs, read, and plot. Typing "snrmin" without arguments removes any lower bound on snr.

COMMAND **ccread**

TYPE **Experiment overview**

SYNTAX **"ccread filename"**

DESCRIPTION

This command reads a correlator control file into memory, thus allowing aedit to compare what "should" be present to what is actually present. Execution of this command is a prerequisite for "psplot" and "psfile".

COMMAND **clear**

TYPE **Action**

SYNTAX **"clear data"**
"clear close"
"clear inputs"
"clear plot"
"clear all"

DESCRIPTION

"Clear data" removes all the data from memory, and returns array space to the system. Since any active plot no longer refers to data in memory after this operation, the plot is rendered inactive.

"Clear close" is analogous to "clear data", but removes only closure data.

"Clear inputs" changes all the values listed by the "inputs" command to their default values. Typically, this means data selection parameters are set to pass all data, and plots revert to self-scaling.

"Clear plot" flushes the current plot, and renders a plot on an interactive device inactive.

"Clear all" simultaneously performs all the above operations.

COMMAND **close**

TYPE **Action**

SYNTAX **close**

DESCRIPTION

Causes all type 2 baseline data in memory to be examined, and closure triangles to be formed. The result is a set of type-3 records in memory, which can then be plotted, edited, filtered and written out to disk just like other data. The close command will refuse to generate closure data unless there are currently no triangle records present.

COMMAND **device**

TYPE **IO control**

SYNTAX **"device postscript"**
"device hpgl"
"device xwindow"
"device name/device"

DESCRIPTION

Sets the device type used for plotting. The available devices are those accessible to the PGPLOT library, which as of February 1994 consisted of tektronix emulators, an Xwindow graphics window, Hewlett-Packard Laserjet printers, and postscript printers.

The "device" command can be invoked in 2 ways. First, there are 4 keywords that are recognized, namely "ppostscript", "lpostscript", "hpgl" and "xwindow". If one of these keywords is specified, aedit will use the corresponding device in a transparent, automatic way. The hardcopy options, "ppostscript", "lpostscript" and "hpgl", send the plot output to whatever printer is specified by the shell script "aedit_plot". The "hpgl" output is portrait, while the postscript output can be either landscape (lpostscript) or portrait (ppostscript). The printing is done immediately, without the need for a "clear plot" command or separate invocation of a printer job.

The second method involves direct access to the PGPLOT device specification mechanism, as described in detail below.

The construction of the argument is in two parts. The first part is the specific name of the output file or device, The second part specifies the type of device. The former can be a standard UNIX filename, such as "plot01.3C345", but subdirectory specifiers (i.e. filenames with "/" in them) are special because PGPLOT is looking for a "/" to separate the two parts of the device specifier. You must "hide" the UNIX "/" characters from PGPLOT by enclosing the filename in double quotes, so that a valid specification for a workstation tektronix emulator might be "'/dev/ttyp2"/te'.

The default filename for interactive devices is the users terminal, whilst for the hardcopy devices, it is "PGPLOT.device". PGPLOT translates filenames to upper case on output.

The second part, the device type, follows a "/", and a complete list of possibilities can be viewed by setting the device equal to "?", the default setting. The names are minimum matchable (e.g. "/te" will work).

COMMAND **edit**

TYPE **Action**

SYNTAX **"edit inputs unflagged"**
"edit inputs all"
"edit inputs"
"edit cursor"
"edit duplicates procdate"
"edit duplicates qcode"
"edit duplicates snr"

"edit parents"
"edit children"
"edit close baselines"
"edit close triangles"
"edit close"

DESCRIPTION

Sets flags in the data records according to a variety of circumstances. These flags can be selectively unset with the "unflag" command.

"Edit inputs" sets a flag bit in each data record for each data selection input parameter which excludes that data point. Thus, a scan may pass the input filter for stations, but fail that for baselines. The baseline bit would be set, but the station bit would not. Any set bit in the flag field causes the scan to be flagged (i.e. it will not be plotted or written to an output file). The "unflagged" qualifier applies the filter only to currently unflagged data. The default "all" qualifier sets flag bits if appropriate even in currently flagged data.

"Edit cursor" enables the cursor on an interactive graphics device upon which data has been displayed using "plot". The user may type any character (except 'x', 'X', 'a', 'A', 'b' or 'B' .. see below) on the keyboard to edit out the point nearest the cursor. The cursor must be inside the border of a plot, and must be twice as close to the target point than any other point for success. Failure to meet these conditions results in an appropriate error message.

Alternatively, the user may define an area on the plot within which all points are to be edited out. This is accomplished by typing 'a' or 'A' to locate the bottom left corner of a rectangle, and 'b' or 'B' to locate the top right corner. Unfortunately, as yet there is no visual indication of the current location of the rectangle. This may be changed in future releases.

On devices which are not capable of erasing points from the screen (e.g. tektronix emulators), the edited points are marked by being overwritten by a solid square.

Do not use the mouse buttons on workstation tektronix emulators - these return multiple characters which may confuse the program. The cursor editing mode is terminated by typing the character 'x' or 'X' on the keyboard. The same may also be true of Xwindow screens.

"Edit duplicates" removes duplicate scans from the database, ignoring flagged scans. The term "duplicate" refers to identical baseline, scan time, frequency code, experiment number and source. The second argument determines which scan aedit will retain. If "procdate" is specified, it will keep the most recent processing. If "qcode" is specified, the "best" quality code scan is kept. If "snr" is specified, the highest snr scan is kept.

WARNING: Since "edit duplicates" ignores flagged scans, unflagging data may generate more duplicates. Similarly, reading in more data may do the same. In such circumstances, the recommended course is to "unflag duplicates" and rerun "edit duplicates".

"Edit duplicates" and "edit cursor" operate only on type 2 data.

"Edit parents" and "edit children" allow you to construct a consistent set of type-0, type-1 and type-2 data, such as may be needed for data export, archiving, and so on. "Edit parents" flags all type-0 and type-1 data records which have no corresponding children (i.e. types 1 or 2 for root records, and type 2 for corel records). "Edit children" removes all "orphan" type 1 and 2 records (i.e. those type-2 records with neither parent root nor corel records, and type-1 records without parent root records).

"Edit close" flags type 2 baseline and type 3 triangle records according to whether or not the two types of data are consistent with each other. The baseline form of the command flags all baseline records which do not appear in any unflagged triangle records. The triangle form of the command flags all triangle records for which all three constituent baseline records are not present and unflagged. Applying both forms (as happens if the second argument is omitted) results in a fully consistent set of baseline and triangle records in memory.

COMMAND **exit**

TYPE **Action**

SYNTAX **"exit" (no arguments)**

DESCRIPTION

Terminates the current aedit session. All data currently in memory is lost. The plot device, if open, is closed and the plot flushed.

COMMAND **experiment**

TYPE **Data selection**

SYNTAX **"experiment 1953"**

DESCRIPTION

Sets the experiment input data selection parameter. Only one experiment number may be specified at one time. Scans which do not belong to the specified experiment number will not pass the filters applied by edit inputs, read, and plot. Typing "experiment" without arguments removes any restriction on experiment number.

COMMAND **fplot**

TYPE **Action**

SYNTAX **"fplot" (no arguments)**

DESCRIPTION

Enables the cursor on an active plot on an interactive graphics device. The user selects a point by positioning the cursor and typing any character except 'x' or 'X', and the program pops up a fringe plot on the screen. This fringe plot can be dismissed with the 'q' key, and the cursor is then ready for the next point.

Note that the "fplot" command works only in an X-windows environment. Also, to display a fringe plot, aedit must be able to locate the type-2 (fringe) file on disk from which the A-file data were generated. By default, it looks in the CORDATA area, but if the DATADIR environment variable is set, it looks there instead.

"Fplot" is terminated by typing an 'x' or 'X'.

COMMAND **fraction**

TYPE **Data selection**

SYNTAX **e.g. "fraction > 8"**
or "fraction <= 60%"

DESCRIPTION

Sets the fraction of the data processed for this scan which will pass the filtering functions applied in read, edit, and plot. The syntax is quite forgiving. The requirements are that there be an inequality operator, possibly followed by an equals sign, followed by a sensible number, possibly followed by a percent sign. If the percent sign is missing, the number is interpreted as tenths of the scheduled data, instead of a percentage. Spaces are irrelevant.

If "fraction" is typed with no arguments, or with just "0" as an argument, all restrictions on the fraction of data processed are removed.

Note that this filter option operates on the value of the last digit in the ESDESP field of the A-file format, which is placed there by FRNGE or fourfit. Before the implementation of baseline-dependent scan lengths in the schedule files, this number was unreliable. Also, being only a single digit, this quantity is only accurate to the nearest 10%, so more precise values entered with the fraction command are rounded off.

COMMAND **frequencies**

TYPE **Data selection**

SYNTAX **"frequencies XS, C"**

DESCRIPTION

Enters a list of allowed frequencies into the inputs. All alphabetic characters are accepted, in any order, lower or upper case, with or without spaces or commas. Duplicate characters are ignored. Scans which involve frequencies not in this list will fail the filter tests applied by edit inputs, read, and plot. Typing "frequencies" without arguments removes any limitations on frequencies.

COMMAND **grid**

TYPE **Plot control**

SYNTAX **"grid n1 n2" (n1, n2 are integers - n1 <= 2, n2 <= 10)**

DESCRIPTION

This sets the parameter which determines how many subplots appear horizontally and vertically on the plotting surface. The default is one in each direction, the maximum is 2 horizontally and 10 vertically. The character size scales with the number of vertical plots to keep things readable.

COMMAND **inputs**

TYPE **Action**

SYNTAX **"inputs plot"**
or "inputs filter"
or just "inputs"

DESCRIPTION

For use in ascii-terminal interface mode only. Places a summary of the current aedit input settings on the screen. The plot and filter options result in a display of only those inputs pertaining to plotting and data filtering respectively, while the default produces a display of all input parameters.

An example is shown below.

```
****************
| AEDIT INPUTS |
****************


DATA FILTER PARAMETERS
----------------------
Timerange:   88124-125700 to  88126-071500
```

```
Procrange:   89119-0223 to  89119-1256
Stations:    ABNT
Baselines:   AB BN TN
Triangles:   No restriction specified
Frequencies: XS
Experiment:  1996
Qcodes:      56789AD
Type:        2
Snrmin:      10
Snrmax:      40
Bis_snrmin:  None specified
Bis_snrmax:  None specified
Sources:     3C345 3C273 OJ287
Length:      30
Fraction:    No restriction specified
Nfreq:       No restriction specified
Outversion:  0 (i.e. same as that read in)


PLOTTING PARAMETERS
-------------------
Axis:        Plot amplitude against scan_time
              (station-based quantities use reference antenna)
Grid:        plot with 2 horizontal and 5 vertical subplots
Y-scale:     Plot between extrema of data
X-scale:     Plot between extrema of data
Scale:       Plot between mbdelay = -20.000000 and 20.000000
Mode:        Nosplit (multiple sources per plot)
Device:      Device for graphics output = xwindow
```

COMMAND **length**

TYPE **Data selection**

SYNTAX **"length 20"**

DESCRIPTION

Sets the minimum scan length in seconds which will pass the filters applied in edit inputs, read, and plot. Typing "length" without arguments removes any limitation on scan length. This data selection parameter applies only to type-2 data.

COMMAND **mode**

TYPE **plot control**

SYNTAX **"mode split"**
or "mode nosplit"

DESCRIPTION

Toggles the setting of the mode parameter, which determines whether or not the data will be split into one plot per source. The default on startup is "nosplit".

COMMAND **nfreq**

TYPE **Data selection**

SYNTAX **e.g. "nfreq >= 8"**
or "nfreq < 2"

DESCRIPTION

Sets the number of frequencies processed for this scan which will pass the filtering functions applied in read, edit, and plot. The syntax is quite forgiving. The requirements are that there be an optional inequality operator, possibly followed by an equals sign, followed by a sensible number. Spaces are irrelevant. If the inequality is omitted, exactly the specified number of frequencies must be present to pass the filters.

If "nfreq" is typed with no arguments, or with just "0" as an argument, all restrictions on the number of frequencies processed are removed.

COMMAND **nobatch**

TYPE **Miscellaneous**

SYNTAX **"nobatch"**

DESCRIPTION

Enables confirmation mechanism, for running interactively (reverses the action of "batch").

COMMAND **outversion**

TYPE **IO control**

SYNTAX **"outversion n", where n is an integer**

DESCRIPTION

This allows the user to override the output format of the A-file when the write command is used. Currently, only versions 1 and 2 are supported. If you specify version 0 (the default), each line will be written individually with the same format as that in which it originated.

Note that writing data out in a different format version number from the one it originated in will generate fields with undefined values in the output. Generally speaking, undefined strings are set to "???", and undefined numerical quantities are set to zero.

COMMAND **parameter**

TYPE **Action**

SYNTAX **"parameter 1 2 3 ..." (non-interactive form)**
"parameter" (interactive form)

DESCRIPTION

This command causes all unedited type 2 data in memory to be treated as the basis for a parameter extraction operation from disk-resident type-2 files. Specified parameters are placed in a special array attached to each type-2 line in memory. The parameter specification is via key numbers. These numbers may be specified either directly on the input line of the parameter command, or in response to a query from the program if no parameter keys are given. In batch mode, aedit assumes that the former mechanism is being used, and the absence of any keys is treated as an error. Once extracted, the parameters may be written to a file of the user's choice, using the pwrite command.

Obviously, aedit cannot extract parameters unless the relevant type-2 files are on the disk. Make sure the DATADIR environment variable is pointing to the correct data area.

Each invocation of the parameter command obliterates all previous parameters extracted for a previous subset of unflagged data lines.

Below is a list of the available parameters, and their index numbers which must be supplied in a space-delimited list. The total number of parameters allowed is currently 32, and each array of parameters (denoted by the parentheses below) counts one for each array element An index number in parentheses indicates that the parameter is already in memory, but can be selected as a parameter for manipulation and output like the others

```
INDEX  PARAMETER NAME                INDEX  PARAMETER NAME
-----  --------------                -----  --------------
 1:    ref_pcal_amp (6)              29:    yperror
 2:    ref_pcal_phase (6)            30:    suppress
 3:    ref_pcal_diff (6)             31:    ppupdate
 4:    ref_pcal_freq (6)             32:    xslip
 5:    ref_pcal_rate                 33:    yslip
 6:    rem_pcal_amp (6)              34:    badsync
 7:    rem_pcal_phase (6)            35:    ref_drive
 8:    rem_pcal_diff (6)             36:    rem_drive
 9:    rem_pcal_freq (6)             (51):  scan_length
10:    rem_pcal_rate                 (52):  scantime
11:    errate_ref_usb (6)            (53):  amplitude
12:    errate_ref_lsb (6)            (54):  snr
13:    errate_rem_usb (6)            (55):  phase
14:    errate_rem_lsb (6)            (56):  resid_sbd
15:    corel_amp (6)                 (57):  resid_mbd
16:    corel_phase (6)               (58):  ambiguity
17:    rate_error                    (59):  resid_rate
18:    mbdelay_error                 (60):  ref_elevation
19:    sbdelay_error                 (61):  rem_elevation
20:    total_phase                   (62):  ref_azimuth
21:    tot_phase_mid                 (63):  rem_azimuth
22:    incoherent_amp                (64):  u
23:    mhz_arcsec_ns                 (65):  v
24:    mhz_arcsec_ew                 (66):  ref_frequency
25:    pcnt_discard                  (67):  total_ec_phase
26:    min_max_ratio                 (68):  total_rate
27:    lo_frequency (6)              (69):  total_mbd
28:    xperror                       (70):  total_sbd-mbd
```

COMMAND **plist**

TYPE **action**

SYNTAX **"plist"**

DESCRIPTION

This command summarizes the state of extracted parameters in memory. Various states can exist, with varying degrees of overlap between flagged and unflagged records, with and without attached extracted parameters. This command is provided to remind the user of the degree to which he/she has confused him/herself.

A more important function is to attach a numerical identifying tag to each extracted parameter present. This tag is then used to identify the parameter to be examined in filtering operations, using the "prange" command, or the parameter to plot, in eth "axis" command. You can see

whether you got the specification correct by using the "inputs" command after attempting to use "prange", or by looking at the axis labels on the plot.

The id tag is the first field in the output of "plist".

COMMAND **plot**

TYPE **Action**

SYNTAX **"plot <y-axis> <x-axis>"**

DESCRIPTION

Initiates plotting of data in memory on a device of the users choice. The data are divided into reasonable logical units (such as stations, baselines, triangles etc.) before plotting. Only one experiment/frequency combination is plotted on any given page, though such a combination may span many pages. If the input parameter "mode" is set to "split", as opposed to "nosplit", a separate set of plots is generated for each source present.

The data are filtered by the input settings before plotting takes place, so you can plot restrictively without having to actually edit the data.

The optional arguments "y-axis" and "x-axis" determine what variables get plotted against each other. If these arguments are omitted, the axis input settings (which can be set either in the previous "plot" command, or in a separate "axis" command) will be used. If only one axis is specified, it is assumed to be the Y axis, and the X axis is set to "scantime". Certain combinations of axes are nonsensical, and are locked out. For a list of available plot axes, see "help axis".

The behaviour of the plot command is controlled by a few other parameters. The "grid" input setting determines how many plots will appear per page in the x and y directions. The "xscale" and "yscale" input parameters allow user-override of the default range of the plots (normally either the natural range of the data, or a fixed range for phase-like quantities). Scan time is handled slightly differently, in that all plots in a frequency/experiment combination are forced to the same start and stop times on the plot, to ensure that plots line up with each other. Manual override of the default is accomplished via the "timerange" input setting.

The "device" input setting determines what plot device will be used for the plots. If you leave this blank, aedit will prompt you with a list of accessible devices. For details, see "help device". Note also that you must issue a "clear plot" command to make sure aedit has finished writing (buffered) information to the plot. This is crucial for hardcopy devices, as the disk file generated by aedit (which must then be manually sent to an appropriate printer) will be incomplete otherwise.

If you are using an interactive device, particularly on an X-window workstation, you will be able to perform point-and-shoot editing, area editing (see "help edit"), and identification and examination of individual data points (see "help fplot" and "help zoom"). To use these features, you must of course have data plotted on the screen.

COMMAND **prange**

TYPE **Data selection**

SYNTAX **e.g. "prange 2 >4.6"**
or "prange 17 <1.5e-12"
or "prange 1 -25 74"

DESCRIPTION

This command sets the input filter for a selected extracted parameter. The parameter is identified by the first argument, which is the identification tag of the parameter reported by the "plist" command. It is thus not easy to run "prange" without first executing "plist". Neither of these commands work, obviously, unless you have already extracted some parameters with the "parameter" command.

The parameter data range of this filter can be specified either with an inequality (no >= or <= because all parameters are floating point quantities internally), or two numerical values, a lower and then an upper limit. To exclude a finite range of values, you must merge two input files, each of which has had one of the inequality limits applied. This limitation will be removed in due course. Thus, the above examples will pass, respectively:

All scans with values of extracted parameter 2 greater than 4.6

All scans with values of extracted parameter 17 less than 1.5e-12

All scans with values of extracted parameter 1 between -25.0 and +74.0

In the same manner as all filter settings in aedit, the data flagging occurs only upon invocation of the "edit inputs" command. In the case of extracted parameters, the filtering during a read operation which normally occurs is suppressed.

COMMAND **procrange**

TYPE **Data selection**

SYNTAX **"procrange yyddd-hhmm yyddd-hhmm"**

DESCRIPTION

Sets the range of procdates outside which data will be rejected by various filter-applying action commands (edit inputs, read). Typing "procrange" without arguments removes any restriction on the procdate range.

COMMAND **psfile**

TYPE **Experiment summary**

SYNTAX **"psfile filename"**

DESCRIPTION

This command is not yet implemented. Sorry.

COMMAND **psplot**

TYPE **Experiment summary**

SYNTAX **"psplot"**

DESCRIPTION

This command takes the data in memory plus the image of a cc file in memory and constructs a 2-D array of quality codes. This is then displayed in an Xwindow PGPLOT window as a (possibly multi-page) colour-coded matrix. Interactive cursor operations are then invoked for data perusal, editing, and fringe plot popups.

The on-screen buttons are self-explanatory. In general, the left mouse button either tags or pops up fringe plots (depending on the setting of the on-screen buttons), while the middle button prints cell identification information in the lower left corner of the window. The right button immediately zaps the data point from the database. Tagged cells are indicated by a small white triangle in the center of the cell. The keystrokes 'a', 'f' and 'x' cause tagging, fringe plot popup, and immediate zapping respectively, regardless of the setting of the on-screen buttons. You can tag many cells at once. To tag an entire baseline, click on the baseline label. To tag an entire scan, click on the scan label. To tag all cells with a particular quality code, click on the quality code key at the bottom of the window. All tagging is done on a toggle basis (i.e. do it again and the tag disappears). Upon exit of psplot, you can zap, write out to disk, or ignore the list of tagged records.

COMMAND **pwrite**

TYPE **Action**

SYNTAX **"pwrite filename"**

DESCRIPTION

Writes all unflagged user-extracted parameter data in memory out to the filename specified in the argument. The data are written out according to the current sort order (as determined by execution of the "sort" command). If the data are not sorted, the output order is the same as the order in which the data were read. You must execute the parameter command before using pwrite. Unflagged data lines which for any reason do not have associated extracted parameters are ignored by pwrite.

The list of user-extracted parameters is preceded by information identifying the baseline, scan and extent number, together with a few other generally useful items (but far less than is present in the A-file format).

COMMAND **qcodes**

TYPE **Data selection**

SYNTAX **"qcodes 5,6,789,DEF"**
or "qcodes 5-9 D-F"
or "qcodes not 0-4 A-C"

DESCRIPTION

Sets the quality code data selection input parameter. Shown in the example are three ways of establishing the quality code filter "56789DEF". You can specify codes directly, in any order, separated by spaces, commas, or nothing at all. You can also specify ranges of quality codes from the sequence "ABCDEF0123456789" by using the construction "2-8". Preceding a specification by the exact string "not" means take all except the specified codes. This information is applied as a filter by edit inputs, read, and plot. Typing "qcodes" with no argument removes any limitation on quality codes.

COMMAND **read**

TYPE **Action**

SYNTAX **"read filename"**

DESCRIPTION

Reads data in from the filename specified in the argument. If enough fields on the line are successfully decoded to identify the parent data file, a data entry is made in memory. If not even enough could be decoded to id the file, the line is skipped. You can read as many files into aedit, one after the other, as you like. "Read" filters the incoming data according to the data selection input parameters.

The unlimited data capacity of aedit is achieved by using dynamic memory allocation inside the "read" function. As more memory is needed, the program obtains it from the system. This memory is released by the command "clear data", or by "exit". The user is informed of memory usage during the reading operation.

COMMAND **reference**

TYPE **Plot control**

SYNTAX **"reference"**

DESCRIPTION

When a station-based quantity, like elevation, is plotted against a baseline-based quantity, like SNR, either the reference or remote elevation must be used. This command specifies that it should be the reference quantity which gets plotted, which is the startup default. There is a corresponding "remote" command.

COMMAND **remote**

TYPE **Plot control**

SYNTAX **"remote"**

DESCRIPTION

When a station-based quantity, like elevation, is plotted against a baseline-based quantity, like SNR, iether the reference or remote elevation must be used. This command specifies that it should be the remote quantity which gets plotted. There is a corresponding "reference" command, the results of which are the startup default.

COMMAND **run**

TYPE **action**

SYNTAX **"run filename"**

DESCRIPTION

Causes the aedit commands in "filename" to be executed, just as if they were typed at the terminal. For obvious reasons, there are a couple of exceptions. Confirmation is no longer requested when using run files, and certain interactive operations (edit cursor, zoom) are disabled. Aedit command files can be nested up to a depth of 10. Any error within a run file causes the execution to abort, and control returns to the terminal, regardless of how deeply the runfiles are nested.

COMMAND **setyear**

TYPE **Action**

SYNTAX **"setyear 1989"**

DESCRIPTION

This command is present only to allow the user to circumvent an unfortunate problem with the A-file format. Some A-files have the year of the scan in field 7, but in others this information is replaced by the number of the parent type-51 HP-1000 extent. Generally, aedit will recognize the latter type of A-file on read, and notify the user that the scan year information is missing from some of the data. In such cases, the year is set to 1980. The recommended course of action is for the user to set the timerange to the offending span in 1980 with all other filters wide open, run "edit inputs" to flag all good data, force the year to the correct value with "setyear", and unflag the good data again. If all data is actually from the same calendar year, the edit and unflag steps are unnecessary - you can run setyear on the whole dataset.

Confusing things could happen if the parent extent number exceeds 80, but this should be almost never.

This command should become obsolete as the move to UNIX proceeds.

COMMAND **snrmax**

TYPE **Data selection**

SYNTAX **"snrmax 20"**

DESCRIPTION

Sets the maximum snr which will pass the filters applied in edit inputs, read, and plot. Typing "snrmax" without arguments removes any upper bound on snr.

COMMAND **snrmin**

TYPE **Data selection**

SYNTAX **"snrmin 20"**

DESCRIPTION

Sets the minimum snr which will pass the filters applied in edit inputs, read, and plot. Typing "snrmin" without arguments removes any lower bound on snr.

COMMAND **sort**

TYPE **Action**

SYNTAX **"sort scantime"**
"sort procdate"
"sort snr"
"sort length"
"sort baseline"
"sort triangle"
"sort frequency"
"sort sourcename"
"sort qcode"
"sort experiment"
"sort rootcode"
"sort family"

DESCRIPTION

Sorts the data in memory according the the value of the field specified in the command. The sort is stable, in that entries in the database which compare equal will retain their original sort order. Thus any combination of sort key priorities can be applied by repeated execution of "sort".

The only consequence of sorting the data is that the output of the "write" command will be sorted. The various "aedit" commands do not care whether the data are sorted or not, and in fact operate on unsorted data whether sort has been executed or not. Flagging and unflagging data does not affect the sort order ... sorting is done on all the data.

The sort keys "snr", "length" and "frequency" have no effect on root or corel data. In addition, the sort keys "baseline" and "qcode" have no effect on root data. The "snr" sort key uses the bispectral snr for triangle records.

Note that reading in additional data destroys any sort order. The effect is the same as issuing the "unsort" command, in that the sort order information for the original dataset is explicitly discarded, and the original sort order restored.

The special key "family" causes the "write" command to write data to disk grouped by root family. Each family has the root record (if present), followed by the corel records (if present), followed by the fringe records (if present).

COMMAND **sources**

TYPE **Data selection**

SYNTAX **"sources 3C345, 3C273, OJ287"**

DESCRIPTION

Specifies a list of sources which will pass the filters in edit inputs, read, and plot. The source names must match those in the data files exactly (including case), with the exception of leading or trailing blanks. Typing "sources" with no arguments removes any restriction on sources.

COMMAND **stations**

TYPE **Data selection**

SYNTAX **"stations ABC D EF"**

DESCRIPTION

Enters a list of allowed stations into the inputs. All alphabetic characters are accepted, in any order, lower or upper case, with or without spaces or commas. Duplicate characters are ignored. Baselines which involve stations not in this list will fail the filter tests applied by edit inputs, read, and plot. Typing "stations" without arguments removes any limitations on stations.

COMMAND **summary**

TYPE **Action**

SYNTAX **"summary"**
"summary 0" or "summary root"
"summary 1" or "summary corel"
"summary 2" or "summary fringe"
"summary 3" or "summary triangle"

DESCRIPTION

Displays a summary of all unflagged data in memory on the terminal. Without arguments, a terse summary of all data of all types is given. If a record type is specified, much more detailed information is provided. An example is given below.

```
                SUMMARY OF UNFLAGGED DATA IN MEMORY
                -----------------------------------

Total number of unflagged fringe records = 6754

Earliest scan:       94-015-183000
Latest scan:         94-016-181505
Earliest procdate:   94-050-1648
Latest procdate:     94-055-1044
Stations present:    DAKLETV
Baselines present:   DA DK DL DE AK AL AE KL KE LE TE AT AV TV EV KT KV DV LV DT LT
Frequencies present: XS
SNR extrema:         0.000  1069.
Experiments present: 2498
Sources present:     0048-097 0059+581 0119+041 0229+131 0454-234 0458-020
        0528+134 0537-441 0552+398 0727-115 0735+178 0804+499 0820+560
        0823+033 0919-260 0954+658 0955+476 1034-293 1044+719 1053+815
        1104-445 1128+385 1219+044 1308+326 1334-127 1357+769 1424-418
        1606+106 1622-253 1726+455 1739+522 1741-038 1749+096 1921-293
        2145+067 2234+282 2255-282 4C39.25 NRAO512 OJ287 OK290
Quality code summary:
        A B C D   E F  0  1 2  3 4  5  6   7   8   9    ?
        0 2 0 137 3 93 88 0 46 5 18 60 144 211 851 5096 0

There are 0 flagged records present
```

COMMAND **timerange**

TYPE **Data selection**

SYNTAX **"timerange yyddd-hhmmss yyddd-hhmmss"**

DESCRIPTION

Sets the range of times outside which data will be rejected by various filter-applying action commands (edit inputs, read). Overrides self-scaling of the time axis on plots. Typing "timerange" without arguments removes any restriction on the timerange.

COMMAND **type**

TYPE **Data selection**

SYNTAX **"type 0 1 2 3"**

DESCRIPTION

Sets the type data selection parameter. Only records of the specified type(s) will pass the data editing functions in "read", "edit" and "plot". The digits 0, 1, 2 and 3 can occur anywhere in the argument. Omission of the arguments implies "type 0123".

COMMAND **unflag**

TYPE **Action**

SYNTAX
"unflag all"
"unflag duplicates"
"unflag cursor"
"unflag qcodes"
"unflag snr"
"unflag bsnr"
"unflag timerange"
"unflag procrange"
"unflag stations"
"unflag baselines"
"unflag triangles"
"unflag experiment"
"unflag frequencies"
"unflag type"
"unflag sources"
"unflag length"
"unflag fraction"
"unflag nfreq"
"unflag parameter"
"unflag parents"
"unflag children"
"unflag nobaselines"
"unflag notriangles"

DESCRIPTION

Unsets flag bits for all scans in memory, according to the argument. For example, "unflag snrmin" unsets all flag bits thoughout memory which were set with an "edit inp" with "snrmin" set higher than the scan snr value. The combination of "edit" and "unflag" allows great control over the flagging status of the data in memory. "Unflag all" removes all flags from the data simultaneously.

COMMAND **unsort**

TYPE **Action**

SYNTAX **"unsort"**

DESCRIPTION

Restores the original sort order of the data (i.e. the order in which the data was read in).

COMMAND **write**

TYPE **Action**

SYNTAX **"write filename"**

DESCRIPTION

Writes all unflagged data in memory out to the filename specified in the argument. The data are written out according to the current sort order (as determined by execution of the "sort" command). If the data are not sorted, the output order is the same as the order in which the data were read. The output A-file format version is controlled by the "outversion" command, and defualts to the same as that read in.

The input filters are ignored on "write". The way to write out selected data is to set the input filters and then run "edit inputs", before using "write".

COMMAND **xscale**

TYPE **Plot control**

SYNTAX **"xscale xmin xmax" (xmin, xmax floating point)**

DESCRIPTION

Sets the minimum and maximum X-axis values, overriding the default, which is 0 to 360 degrees for phase quantities, or the data range plus 10% at each end for other quantities. Scale with no arguments restores the default. When the X-axis is multiband delay, the scale is automatically set to +/- half the multiband delay ambiguity. This can be overridden by an explicit "xscale" command. If the axis is scantime, the "timerange" settings determine the axis extrema.

Points which fall outside the scale limits are not plotted, and a warning message is issued to alert the user as to how many points were omitted.

COMMAND **yscale**

TYPE **Plot control**

SYNTAX **"yscale ymin ymax" (ymin, ymax floating point)**

DESCRIPTION

Sets the minimum and maximum Y-axis values, overriding the default, which is 0 to 360 degrees for phase quantities, or the data range plus 10% at each end for other quantities. Scale with no arguments restores the default. When the Y-axis is multiband delay, the scale is automatically set to +/- half the multiband delay ambiguity This can be overridden by an explicit "yscale" command. If the axis is scantime, the "timerange" settings determine the axis extrema.

Points which fall outside the scale limits are not plotted, and a warning message is issued to alert the user as to how many points were omitted.

COMMAND **zoom**

TYPE **Action**

SYNTAX **"zoom" (no arguments)**

DESCRIPTION

Enables the cursor on an active plot on an interactive graphics device. The user selects a point by positioning the cursor and typing any character except 'x' or 'X', and the program displays detailed information about that point on the terminal. The only information displayed is that resident in memory (i.e. A-file information). If you want more detail, make sure the binary data files are on disk, set the DATADIR variable appropriately, and use the "fplot" command to pop up a fringe plot on your Xwindow screen.

"Zoom" is terminated by typing an 'x' or 'X'.