AF2_CONFIGURE USERS GUIDE

Royal Greenwich Observatory
Madingley Road,
Cambridge CB3 0EZ

J.R. Lewis
September 1996

Contents

1 Introduction

In this document we describe the use of the program **af2_configure**, which is designed to create the mapping between the objects that will be observed during a particular **wyffos** exposure and the fibres that will observe them. In its most general mode it will take an input file with the coordinates (α, δ) of the objects, create a fibre to object mapping using one of two currently available placement algorithms, and then allow the user to edit the placement interactively. Three files are required for input to the program: a fibre information file, a telescope information file and an object field file. In general only the last of these will be required of the user. In section 2 we describe all three of these files and give an accounting of the required information for the input object field file. During a run the user will be prompted for some information, which will be explained in Section 3. A sample run is also included. Section 4 explains how to use the interactive editor and describes each of the available editing options. The available placement algorithms are described in section 5. If you have a done a configuration on a field and you would like a hard copy of that configuration without having to go back into **af2_configure**, then the application **af2_hcopy** is available and is described in Section 6. Section 7 gives instructions on how to obtain and install the package. Finally, in the appendices we list some common errors and some programming notes.

2 The input files

2.1 The fibre information file

One thing which is common to all autofib-2 bundles is that there are two basic type of fibres, program or fiducial. The former are used for the program objects and the latter for fiducial or guide objects. Both of these types of fibres will be assigned a status according to one of these criteria:

missing The fibre is physically missing

disabled The fibre has been disabled for some reason (perhaps it is broken).

dead The fibre has very low transmission.

ok The fibre can be used for placement.

The fibre status will be decided by the micro which runs autofib-2 and the results will be put to the 'fibre information file'. This file also contains other information such as the park position of each fibre and their current x,y positions. The name of this file is constructed by adding the name of the bundle to the string '_fibres.dat'. Thus the fibre information file for the 'large' bundle will be called 'large_fibres.dat'. A copy of this file will be provided with the executable image of **af2_configure**. It may be the case however, that the installed version is rather out of date. In that situation it is possible to obtain a newer version and tell **af2_configure** how to find it. Section 3 explains how to do this.

2.2 The telescope information file

The scale and distortion of the field of the telescope is stored in the telescope information file, called 'wht_prime.dat'. As with the fibre information file it is provided with **af2_configure**, but may well go out of date. If this is the case, a better version can be obtained and used. Again see Section 3 to find out how this is done.

2.3 The input field file

In general the only input file required of the user will be the file which contains the coordinates of the objects to be observed. This we will call the *field* file from now on. The field file has two sections – the header and the object information. Comment lines may appear anywhere within the field file. These have an asterisk in the first column and will be ignored by **af2_configure**. The field file can be called anything you want, so long as it has the extension '.fld'. Note that **af2_configure** is *always* case sensitive in the specification of file names. Therefore the field file extension must be in lower case.

There are seven header items, all of which you *must* specify in each field file. (NB: The header items can appear in whatever order you wish, but they must *all* be there once and only once and they must appear in the field file *before* the first line of object information.) The header items are as follows:

centre The RA and DEC of the centre of the observation field. The hours and minutes columns of the RA and the degrees and minutes columns of the DEC should be expressed as integers. The seconds columns of both RA and DEC should be expressed as reals. Example:

    CENTRE 12 34 15.3 -00 30 17.4

equinox The equinox of the coordinates. Note that the equinox of the field centre *must* be the same as that for the objects! Example:

    EQUINOX 2000.0

utdate The date of the observation expressed as the year and a fraction. For example, if observations are being done on 30 June 1995, the value of **utdate** would be 1995.50. Example:

    UTDATE 1995.89

ha The hour angle of the observation ($T - \alpha$, where T is the local sidereal time). This is vital in order to calculate the refraction correction properly. If a configuration is done with an hour angle which is vastly different from that which will be used at the telescope, **af2_configure** may not pick up collisions between fibres which might occur. This may mean wasted time at the telescope while another configuration file is created. If configurations are being done in advance, it may well be worthwhile doing several runs with different values of the hour angle and then at the telescope choosing the one appropriate for the time of observation. Example:

    HA 0.5

target The name of the field. Each of your fields should have a unique name. This will be important at the data reduction stage. Example:

TARGET Field1 in SGP

skypa The position angle of the spectrograph on the sky. This should be expressed in degrees. It can sometimes be useful to change the value of **skypa** in order to get a better solution, especially with regard to fiducial objects. If **af2_configure** is used in angle scanning mode (see Section 3), then this value will be ignored. Example:

SKYPA 90.0

fibres The name of the fibre bundle to be used in the observations. The only option which is currently available is 'LARGE'. Example:

FIBRES LARGE

All of the information for each object should all appear on a single line. See the sample field file below for examples. The data required for each object is:

name (Column number 1) This should be any alphanumeric string representation of the object name. Note that there should be **no** spaces embedded in the name! Perhaps use underscores instead.

ra (Column number 2–4) This is the RA of the object. The hours and minutes must be expressed as integers; the seconds should be expressed as a real.

dec (Column number 5-7) This is the DEC of the object. The degrees and minutes must be expressed as integers; the seconds should be expressed as real.

type (Column number 8) This is the object type and can take one of three values:

P – Program object

F – Fiducial object

S – Dedicated sky patch

Any other value will result in an error and the object will be ignored. Note that object types are *all* upper case.

weight (Column number 9) The user can weight the objects in order to increase the chance that a particular object will be allocated. The weighting can be according to any arbitrary scheme so long as the value is expressed as a real number which is greater than or equal to zero. Weights will be interpreted in the sense that a higher numerical value corresponds to a higher weight. Avoid weights of zero unless that is exactly what you mean.

magnitude (Column number 10) This is an estimate of the magnitude of the object in any band you wish and is only optional. The value is read as a real number.

anything else You can fill out the rest of the line with other information about the object. Make sure you prefix these comments with an asterisk. Otherwise **af2_configure** may think that the first word of your comments are in fact the object's magnitude! This information will for the most part be ignored by **af2_configure**, but it may provide some useful notes for you.

Below is a small example field file.

```
TARGET Example field file
UTDATE 1995.6
EQUINOX 1950.0
SKYPA 0.0
FIBRES LARGE
CENTRE 22 33 00.0 1 15 00.0
HA 1.0
S115 22 32 58.72 0 53 43.3 F 1 12.41 *
S2 22 31 49.84 1 44 54.3 F 1 12.46 *
S49 22 34 29.97 1 22 31.0 F 1 12.47 *
S10 22 33 15.48 1 40 30.1 F 1 12.50 *
S74 22 34 43.05 1 18 14.0 P 0.5 15.97 *
S52 22 32 38.00 1 22 56.9 P 0.5 15.98 *
S156 22 33 58.15 1 5 16.1 P 0.5 15.98 *
S151 22 33 28.17 1 6 11.9 P 1 16.01 * carbon star??
S103 22 32 8.10 1 14 38.8 P 1 16.05 *
S99 22 33 2.83 1 14 55.1 P 1 16.14 *
S207 22 31 24.14 0 50 58.7 S 1 *
S175 22 33 57.09 0 59 56.3 S 1 *
S122 22 32 27.50 1 12 17.0 S 1 *
```

3 Running af2_configure

3.1 Prompted information

If **af2_configure** has been installed at your site, then you should be able to access it by just typing '**af2_configure**' at the UNIX prompt. If this fails then contact your system manager. It may be that you need to put something in your default path in order to access the binaries.

As with most programs, **af2_configure** requires a certain amount of information from the user. In this section we give a listing of all of the prompts which you might encounter during a run. These will be roughly in the same order as they occur during a run. The values in parenthesis are the default values. Note that not all prompts will have defaults. If you wish to accept the default, then you can simply hit the carriage return button. (In what follows below, some prompts have parenthesis with nothing in them. This indicates an prompt which has a default which is decided at run time.) If you want to get out of **af2_configure** gracefully, you may type '!!' at any of the prompts and the program will abort. If you answer any of the questions in a way which **af2_configure** can't understand (*e.g.* you answer a yes/no question with a number) then you will be reprompted.

Define WHT_AF2DATA_DIR ( ) In **af2_configure** the fibre information file and the telescope information file are sought in a directory which is determined by the environment variable 'WHT_AF2DATA_DIR'. If you have set this, *e.g.*

```
% setenv WHT_AF2DATA_DIR /data/disk3/jrl/
```

then you will just be informed of its value. If you haven't set this up in advance, then this prompt will appear with the default being the directory where the binary for **af2_**

configure is located. In most cases you can just hit return and accept the default value. If you have new versions of the fibre information file and the telescope information file, then you'll need either to answer this prompt with the directory which holds these files, or define the directory before invoking **af2_configure**, as per above.

Do initial placement by scanning SKYPA? (no) **af2_configure** can be run in an angle scanning mode which allows you to sample a number of different values of the sky position angle (**skypa** in the field file) in order to try and get the best possible orientation.

Input field file This is the name of the field file – that is the file with the RA and DECs of the program objects described in section 2.3. In specifying the file, it is not necessary to include the '.fld' file extension.

Do you want verbose output? (no) If you want some extra information as the program is running, you should answer 'yes' here.

Use old configuration file? (no) If you have done a configuration on this field before, then you may use the old configuration file as a starting point for this run. (For a discussion of configuration files, see section 3.3.) This can be quite useful if you want to break off a run and pick it up again later. It can also be useful if you don't like a certain configuration and all you want to do is to edit it. Note that you cannot use an old configuration file in angle scanning mode.

Which version do you want to use ( ) If you have decided you want to use an old configuration file, you will be prompted for a version number. The default in the prompt will be the most recent version number. If you only have one old version, then you won't be asked this question.

How many position angles to try? (10) This is the maximum number of position angles to scan in angle scanning mode.

Which statistical measure to use? (2) While in angle scanning mode, **af2_configure** decides which is the best scan by either (1) seeing which scan results in the most objects allocated or (2) seeing which scan results in the highest total weight. (For uniformly weighted objects, these are obviously equivallent.)

Buffer factor for fibre dimensions (1.00) This is a factor by which the tube width and the button dimensions will be multiplied in order to increase their sizes artificially. This is quite useful if you are worried that possible fibre collisions will cause **af2_configure** to generate illegal configurations. For example, if the buffer factor is 1.05, then the fibre dimensions will be increased by 5%. The value can not be less than 1.0.

Which fibre placement algorithm? (2) This is the prompt you will get for the fibre placement algorithm when using **af2_configure** in its normal mode (not in angle scanning mode). See Section 5 for a description of the available algorithms.

Which algorithm to use during the scanning? (1) This is the prompt you will get when in angle scanning mode for the algorithm to be used during the scanning.

Which algorithm to use for final placement? (2) This is the prompt you will get when in angle scanning mode for the algorithm to be used after the scanning when you have decided which is the best position angle.

Iterate the placement? (yes) Once the placement has been done either by using one of the placement algorithms, or by reading the initial placement from an old configuration file, then you may iterate the configuration in an attempt to increase both the number of objects placed and the total weight of the objects placed. This is usually helpful and is recommended.

Plot configuration and allow for editing? (yes) If you want to edit the configuration, or if you just want to have a look at it, then answer 'yes' here.

Show whole plate? (no) If you are going to plot the configuration you may display the entire fibre plate if you so wish. If the whole fibre plate is displayed it can make seeing the objects and the fibres a bit difficult. You do have the option under these circumstances to zoom, though. There aren't many reasons for wanting to see the whole plate, so if you answer 'no' here, the maximum view you will have is 1.1 times the park radius, which certainly makes for a better view.

Graphics device/type (? to see list, default /NULL) This is the prompt for a graphics device direct from PGPLOT. If you are not sure, then type '?' for a list of the available devices at your site. (/xw will usually get the xwindows server.)

Number of fiducials required for acquire ( ) This is the number of fiducials to put at the end of the output configuration file. This is for the use of the routine **af2_acquire**, which is run at the telescope. The default value will be the total number of fiducials which were read in from the field file. If you don't know or don't care, then just use the default.

Generate an x,y file? (no) Answering 'yes' here will cause a file with the extension '.fib' to be generated. In it will be the x,y positions of the fibres. This may be useful in comparing the results of this program with that of **af2_setup**.

Generate a new field file? (no) You have the option to create a new field file which will be exactly the same as the current one with two exceptions: (1) all of the objects which have been configured on this particular run will be zero weighted and (2) the field name will be altered slightly. The former is so that you can run a second configuration on the same field and you will preferentially allocate the objects which didn't get allocated the first time around. The latter ensures you don't have problems at the data reduction stage.

New field file name If you have opted to create a new field file as described in the previous item, then this is the name of the new field file. Omit the '.fld' extension. Also, make sure that it has a different name from the original field file. Otherwise the latter may get clobbered by the new file.

3.2 Sample runs of af2_configure

Below is a sample run of **af2_configure**in its ordinary mode. Some comments on the run:

1. The user has let the value of WHT_AF2DATA_DIR default to the location of the **af2_configure** binary. If new versions of wht_prime.dat and large_fibres.dat were available, then the user should request that WHT_AF2DATA_DIR be set to the directory where they are stored.

2. By answering 'no' here the user will be running **af2_configure** in its usual mode (*i.e.*not in angle scanning mode).
3. As is seen on the following line, the input file is called 'f891.fld'. Note that the user has omitted the file extension; typing the extension isn't necessary.
4. The user has opted for verbose output. Much of what you'll see following this line wouldn't be printed out if **af2_configure** were being used in a quiet mode. For the sake of brevity much of the repetitive information which the user would have received at the terminal during this run has been edited out.
5. The fact that the user was prompted for this means that there already exists at least one configuration for this field in the current working directory (the first such file would be called 'f891.cfg.1'). The fact that he has answered 'no' to this question means that a completely new configuration will be generated now. If he had answered 'yes', then a version number would have been prompted for and the program would have skipped to reference mark 7 and then to 10.
6. The program has read the field file. A number of the objects in the field file are outside the field radius and will be skipped. These are denoted here.
7. This is an accounting of the various types of objects which have been read in from the field file.
8. The user has decided to use the fibre dimensions as they have been read from the fibre information file. If there was a certain amount of paranoia about possible collisions causing illegal configurations, then a value of 1.05 would have been reasonable.
9. The SLOW algorithm will be used. As previously mentioned, this is the algorithm which tries to maximise the total sum of weights of the objects placed.
10. This is an accounting of how well the placement algorithm did. Note that no sky patches were placed. This is not necessarily a problem since there were no sky patches read in from the field file (see reference label 6).
11. The user has decided to iterate the initial placement. As can be seen from the progress report it has resulted in the allocation of more objects and an increase in the sum of the weights.
12. Both placement algorithms will initially ignore objects with a zero weight. If there are any fibres left over after the initial placement, then **af2_configure** allows the 'old' algorithm to try and place these spare fibres on objects which have a zero weight (*i.e.* something is better than nothing). In this instance, no extra fibres were placed.
13. The user has opted to edit the placement. The maximum radius of the display will be a little greater than the park radius and the PGPLOT device will be '`/xw`', which is an X-windows device. If you are not sure what to type here, you can type '?' for a list of available devices on your system.
14. After editing the user has opted to add 10 fiducial objects to the output configure file for the use of **af2_acquire**.

15. The user has decided not to write out a file with the x,y coordinates of the fibres, nor will a new field file be written out.

```
Define WHT_AF2DATA_DIR (/home/jrl/disfigure/dev) >                     (1)
Taking standard telescope and fibre info from: /home/jrl/disfigure/dev
Do initial placement by scanning SKYPA? (no) > no                      (2)
Input field file > f891                                                (3)
Input field file is: f891.fld
Do you want verbose output? (no) > yes                                 (4)
Use an old configuration file? (no) > no                               (5)

Reading through field data file f891.fld


Field label is F891 GALAXY FIELD
UT date of observation is 1995.6
Equinox is 1950.0
Sky PA is 0
Fibre bundle is LARGE
Field centre is 22 33 00 1 15 00
Hour angle is 0


Reading in object data
Object outside field radius  104608.22  -33698.21                      (6)
Object outside field radius   78577.63 -105197.33
Object outside field radius   96937.29   89367.50
Object outside field radius   61893.86  105188.60
Object outside field radius  -24283.27  104391.30
Object outside field radius  -91821.71   95333.59
Object outside field radius   84109.77  -84574.93
Object outside field radius   74866.70   73593.21
.
.    (lines edited out)
.
Object outside field radius  -86203.52   77466.41
Object outside field radius   80513.41   87847.52
Object outside field radius   46928.51 -100869.52

 21 fiducial objects                                                   (7)
162 program objects
  0 sky positions
 38 objects outside of field

Buffer factor for fibre dimensions (1.00) >                            (8)
Trying to configure 183 objects and 136 pivots

Which fibre placement algorithm?
```

```
(1) The FAST algorithm
(2) The SLOW algorithm
Choose one (2) >                                                      (9)
Looking for all possible object-fibre combinations...
   860 possible object-fibre combinations found
Now calculating badness index for each combination
Sorting by badness
Placing the fibres now...
putting fibre 125 on object  33 S34
putting fibre  20 on object 110 S77
putting fibre  26 on object  59 S58
putting fibre  29 on object  82 S56
putting fibre  44 on object  43 S122
putting fibre  12 on object 102 S73
putting fibre 126 on object 123 S96
putting fibre  18 on object 116 S39
putting fibre  11 on object 118 S13
putting fibre  37 on object  66 S111
putting fibre  39 on object 124 S129
putting fibre 128 on object 130 S25
putting fibre 121 on object 120 S20
putting fibre 119 on object  61 S28
putting fibre  99 on object 111 S115
.
.           (lines edited out)
.
putting fibre 115 on object 148 S60
putting fibre  85 on object 173 S201
Looking for all possible object-fibre combinations...
     2 possible object-fibre combinations found
Now calculating badness index for each combination
Sorting by badness
Placing the fibres now...
putting fibre  40 on object 104 S139
Looking for all possible object-fibre combinations...
     0 possible object-fibre combinations found

Total number of objects allocated:  86                                (10)
Total sum of weights of allocated objects:   75.5
Total number of sky patches allocated:   0

 80 program objects placed
  6 fiducial objects placed
  0 sky fibres placed
  0 placed by cursor only
Iterate the placement? (yes) > yes                                    (11)
Current sum of weights   75.5
Iteration:   1, number of attempted changes:  7
```

```
   Sum of weights now:   79.0
Iteration:   2, number of attempted changes:  0
   Sum of weights now:   79.0
Final sum of weights is:   79.0
 88 program objects placed
  5 fiducial objects placed
  0 sky fibres placed
  0 placed by cursor only

Trying to place spare fibres on objects with weight <= 0.0            (12)
 88 program objects placed
  5 fiducial objects placed
  0 sky fibres placed
  0 placed by cursor only
Plot configuration to allow for editing? (yes) > yes                  (13)
Show whole plate? (no) > no
 Graphics device/type (? to see list, default /NULL): /xw
.
.     (fibre plot is displayed and the user edits the configuration)
.
Number of fiducials required for acquire? (21) > 10                   (14)
 88 program objects placed
  5 fiducial objects placed
  0 sky fibres placed
  0 placed by cursor only
Generate an (x,y) file? (no) > no                                     (15)
Generate a new field file? (no) > no
```

Here is another sample run. This is **af2_configure** in angle scanning mode. The comments below refer to items which differ from the previous run.

1. The user has opted to use angle scanning mode
2. There will be four scans (*i.e.*at position angles 0, 90, 180 and 270).
3. The best scan will be judged to be the one which maximises the sum of the weights of the objects allocated.
4. During the scanning of the four position angles, the initial placement will be done by the FAST algorithm.
5. Once it has been decided which is the best position angle, the the SLOW algorithm will be used to do the final placement.
6. These are the running results for each scan.
7. The best result was achieved with a position angle of 0.
8. This is the result of using the SLOW algorithm for the final placement at position angle 0 degrees.

9. The user has opted to iterate the placement as before. The rest of the run is exactly the same as running **af2_configure** in normal mode.

```
Define WHT_AF2DATA_DIR (/home/jrl/disfigure/dev) >
Taking standard telescope and fibre info from: /home/jrl/disfigure/dev
Do initial placement by scanning SKYPA? (no) > yes                    (1)
Input field file > f891
Input field file is: f891.fld
Do you want verbose output? (no) >
How many position angles to try? (10) > 4                             (2)

Which statistical measure to use?
(1) The total number of fibres placed
(2) The total weight of the fibres placed
Choose one (2) >                                                      (3)

Which algorithm to use during the scanning
(1) The FAST algorithm
(2) The SLOW algorithm
Choose one (1) >                                                      (4)

Which algorithm to use for final placement
(1) The FAST algorithm
(2) The SLOW algorithm
Choose one (2) >                                                      (5)
Buffer factor for fibre dimensions (1.00) >

Field label is F891 GALAXY FIELD
UT date of observation is 1995.6
Equinox is 1950.0
Sky PA is     0.00
Fibre bundle is LARGE
Field centre is 22 33 00 1 15 00
Hour angle is 0


Results for scan:  1, position angle:    0.00                         (6)
Total number of objects allocated:  91
Total sum of weights of allocated objects:   71.0
Total number of sky patches allocated:   0


Results for scan:  2, position angle:   90.00
Total number of objects allocated:  88
Total sum of weights of allocated objects:   65.5
Total number of sky patches allocated:   0
```

```
Results for scan:  3, position angle:  180.00
Total number of objects allocated:  92
Total sum of weights of allocated objects:   70.2
Total number of sky patches allocated:   0


Results for scan:  4, position angle:  270.00
Total number of objects allocated:  90
Total sum of weights of allocated objects:   70.5
Total number of sky patches allocated:   0


Best result was:   71.00 for position angle    0.00                  (7)

Final result for position angle:    0.00                               (8)
Total number of objects allocated:  92
Total sum of weights of allocated objects:   83.2
Total number of sky patches allocated:   0

Iterate the placement? (yes) >                                         (9)
Current sum of weights   83.2
Final sum of weights is:   86.0

Trying to place spare fibres on objects with weight <= 0.0
 90 program objects placed
  6 fiducial objects placed
  0 sky fibres placed
  0 placed by cursor only
Plot configuration to allow for editing? (yes) > no
Number of fiducials required for acquire? (21) > 10
 90 program objects placed
  6 fiducial objects placed
  0 sky fibres placed
  0 placed by cursor only
Generate an (x,y) file? (no) >
Generate a new field file? (no) >
```

3.3 Output files

The output from **af2_configure** will be some or all of the following files:

Configuration file This is the main output file and so long as the program completes successfully you will always get one. It is used at the telescope as input to **af2_setup** which is ultimately responsible for moving the fibres to the correct positions. This file will be called by the same root name as the input field file, but with the extension '.cfg' and a version number added. For example if the input field file were called 'm51.fld', the first output configuration file would be called 'm51.cfg.1'. The contents of the configuration file start with a listing of the telescope information file contents. This was done so that there would be no doubt as to which version of the telescope information file was used for any given configuration. The next seven lines are the header from the field file. After the header is a line for each fibre. The first column is the fibre number. This is followed by the information (name, α, δ, *etc.*) on the object to which that fibre has been allocated. Fibres which have not been allocated have a one word description for the reason they are so, *e.g.* 'PARKED' or 'DISABLED'. Fibres which have been placed by cursor positions in the editor will have an object name 'PLACE_XY'. Following that will be the α and δ which **af2_configure** has calculated for the location and the original x,y coordinates. Note that the x,y coordinates for objects are *not* included as they will be recalculated by **af2_setup** at the telescope. That program will also recheck the configuration for fibre collisions. At the end of the configuration file is a list of fiducial objects which can be used by the program **af2_acquire** for locating and centring the region. These objects all have fibre numbers greater than 900 and are ignored by **af2_setup**. Editing this file is *strongly* discouraged.

Log file This is a log of the information read from the field file plus any comments which **af2_configure** may have about it. As with the configuration file above it will have be same root name as the input field name, but with the extension '.log' and a version number. Thus for an input field file called 'm51.fld' the first log file will be called 'm51.log.1'. This file may be useful for you if **af2_configure** has signalled an error in your input file (fatal or not). This may include files falling outside of the field radius or errors in the format of the input file. It is not used as input for any other program so you are free to do with it whatever you like.

x,y file This is an optional file which will be generated if the user requests it as in item 14 in the first example of the previous section. As with the previous output files, it will be called by the same root name as the input field file, but with the extension '.fib' and a version number. The first four lines of the file will be header information. Each of the following lines will have a fibre number followed by the x,y coordinates assigned to that fibre by **af2_configure**, irrespective of whether the fibre has been allocated or not.

Revised field file When running **af2_configure** in angle scanning mode you will almost inevitably create a configuration with a sky position angle which is different from that in the input field file. For that reason, **af2_configure** will silently rename the original field file (by appending the suffix **OLD** to the basename) and create a revised field file which is in every way the same as the original with the exception of the value of **SKYPA**. This will then allow you to use **af2_configure** a second time in normal mode in the event that all you want to do is to edit the configuration. (Remember, the header of the field file and the header of the configuration file *must* match if you are using an old configuration file). You only get this revised field file on angle scanning mode!

New field file This is also an output file which is generated by request. The answer which the user gives should be the root name of the new file. The extension '.fld' will be appended automatically. Note that version numbers are not supported for field files. The contents of the file will be almost the same as that of the input field file except that the NAME in the header will be slightly different (to avoid complications at the data reduction stage, all field should have a unique name), and all of the objects which were allocated on this particular run will have a weight of zero.

Any PGPLOT hardcopy files You'll get these if you request hard copies during an editing session. The file names will depend on the format you request. Default names are hard-coded into PGPLOT. Note that it is possible to request a file name at the same time you are requesting the device type. For example when PGPLOT requests a hard copy device, you could just answer: '/ps' which will give you an output postscript file called 'pgplot.ps'. If you would rather have an output file called 'george.ps' then you can do so by answering 'george.ps/ps' to the PGPLOT hard device prompt.

4 Editing a configuration

At some stage during a configuration you will probably want to see and/or edit the placement. This is done in the interactive editor which appears if you request it in step 12 of the first example in the previous section. For various reasons it is recommended that you use a device with some sort of colour capabilities like a colour graphics X-terminal or Sun workstation.

When the plot is first displayed you should see markers denoting the locations of all of your objects and drawings of each of the fibre tubes and buttons. Tables 1 and 2 explain the colour coding for each.

Table 1: Colour scheme for objects

Object type	**Marker type**	**Marker colour**
program	plus sign	red
sky patch	open circle	magenta
fiducial	asterisk	light blue

Table 2: Colour scheme for fibres

Fibre type	**Colour**
good program	green
good fiducial	blue
disabled program	red
disabled fiducial	magenta
dead program	red+magenta
dead fiducial	blue+magenta

In Table 3 we give a listing of the available edit options in alphabetic order. In what follows we group these options into like actions and give a full description of each. Note that although these options appear in the table all in upper case, the editing commands are not in fact case dependent.

4.1 Getting information

The following options are useful for getting information about fibres, objects and the configuration itself. The information will appear in the message window of the plot.

Table 3: Available editing options

Option	**Action**
#	Type fibre no. and get information on it
A	Undo the most recent undo action
C	Move indicated fibre to cursor location
E	Exit from the editor
F	Get information on the indicated fibre
H	Make a hard copy of the current plot
I	Get information on the indicated object
M	Move the indicated fibre to an object
O	Zoom the plot outwards
P	Park the indicated fibre
R	Refresh the plot
S	Give a status report on the placement
U	Undo the most recent action
X	Zoom the plot outwards to the allowed maximum
Y	Restart the editing session
Z	Zoom the plot inwards

This gives exactly the same information is the **F** command below. The only difference is that after typing the '#' keystroke, the number of the fibre for which information is sought needs to be typed and followed by a carriage return. This therefore allows the user to get information about a given fibre without the need to know exactly where it is on the plot

F Information about the fibre whose button head is closest to the current cursor position is displayed. This includes the fibre number, its type and its current allocation state. The latter includes information on the object the fibre may be covering.

I This gives information about the object which is nearest the current cursor position. This includes a fibre number if this particular object has been allocated.

S The user will receive report of the status of the current allocation. That includes an itemised list of the number of objects covered (separated according to object type) and the sum of the weights of the allocated objects.

4.2 Moving a fibre

The following options allow the user to move a fibre in one way or another. Note that you will not be allowed to move a disabled or dead fibre, nor are you allowed to move a fibre outside of the park circle. In all of these there are certain rules which may prevent you from moving a fibre the way that you would like (*e.g.* no two fibres should cross or touch). If you are forbidden from moving a fibre in a particular way, the reason should appear in red in the message box.

C This is a two-keystroke operation. Pointing the cursor at a fibre head and typing **C** will cause the fibre to be highlighted. Then pointing the cursor at a desired location will cause the highlighted fibre to be moved to that location. Note the second keystroke needn't be a '**C**'.

M This is also a two-keystroke operation. Pointing the cursor at a fibre head and typing **M** will cause the fibre to be highlighted. Then pointing the cursor at an object will cause the highlighted fibre to be moved to that object. The second keystroke needn't be an '**M**'.

P This causes the fibre whose button head is nearest to the current cursor position to be parked.

4.3 Zooming in or out

The following options allow you to change the scale of the current plot (*i.e.* zoom). You cannot zoom out any further than the maximum radius. If you have opted to show the whole plate, then the maximum radius is the pivot distance. If you have not opted to show the whole plate, then the maximum radius is approximately 1.1 times the park radius. Note that for the **Z** and **O** options, the expansion or contraction of the field is done with respect to the location of the cursor. Thus you will be forbidden from user either of these commands with the cursor pointed somewhere outside the plot window,

O Zoom outwards by a factor of 2 in each direction.

X Zoom outwards to the maximum size allowed.

Z Zoom inwards by a factor of 2 in each direction.

4.4 Undoing your actions

af2_configure keeps a buffer with the most recent editing actions which you have performed during the current editing session. It is therefore possible to reverse some of your editing decisions. There is a limit to the size of the buffer which stores the editing history (currently 40 actions, but that can be changed by the person who builds the program) so you may not be able to backtrack as far as you would like. Note that once you leave the editor, all knowledge of the editing session is lost and you will not be able to undo your actions with these keystrokes during another run of **af2_configure**.

A If you have undone an action (keystroke **U**) and you are sorry you did so, then you can recover the situation by 'undoing the undo'. This option can only be used after having used the **U** option at least once and can be used to undo a whole series of **U** actions.

U Undo the most recent action.

Y If you are unhappy with your entire editing session, this option allows you to restore the configuration to the state it was before you started editing.

4.5 Other options

Here are some options which don't seem to fit into any of the other categories.

E Exit from the editor.

H Create a hard copy of the current plot. You will be asked for a PGPLOT hardcopy device (again you can type '?' for a listing of available devices). Once the file has been written, the plot will be redrawn on your screen.

R **af2_configure** attempts to redraw objects which would otherwise be erased when fibres are moved. Sometimes bits of object markers don't get redrawn and the park circle gets partially erased. You can use this option to refresh the plot and redraw anything which may be missing.

5 The placement algorithms

In this section we give a description of the two placement algorithms which are currently available in **af2_configure**. We also try to stress the benefits and the weaknesses of each method. In the original version of **af2_configure** and this document we referred to these algorithms as the 'old' and the 'new' algorithms. This wasn't very helpful. So in this version we have renamed them as the FAST and the SLOW algorithms. (This is slightly more informative but doesn't gain much in terms of imagination – sorry.)

5.1 The FAST algorithm

The FAST algorithm works on the principle of trying to place a fibre on an object in such a way that the placed fibre blocks as small an area of the rest of the field as possible, thus leaving a larger area of the field available for other fibres. This is done by starting with the last usable fibre in the bundle (in the numerical sense) – this should appear on the top of the fibre plot in the editing mode. The program works out which object can be reached by that fibre and will cause the fibre to extend to the maximum angular distance (as measured from its park position) in the anti-clockwise sense, without exceeding a preset limit imposed in the fibre information file. Again, we are restricted by various rules which don't allow fibres to cross or touch. Once a fibre has been considered (and placed on an object, if possible) the next fibre to be examined is a quarter of the way around the pivot circle. We keep jumping around the circle in quarters until all fibres have at least been considered.

The major advantage of this method is its speed. Even fields which are strongly clustered are dealt with very rapidly. Its main disadvantage is that it takes no account of the weight placed on the object (with the exception that it ignores objects with a zero weight). This means that although you may place a large number of objects, they may not necessarily be the ones you most want.

5.2 The SLOW algorithm

The SLOW algorithm attempts to maximise both the total number of fibres placed and the sum of the weight of the allocated objects. It does this by first finding all of the possible fibre-object combinations which don't break any of the placement rules. It then looks at each combination one by one and assess how many other combinations this particular one will block. A penalty function for the combination is computed based on the number of blockages, the weight of each object blocked and the weight of the object in the combination. Thus the penalty function tries to balance the blockage of some objects with the need to allocate the current object. Once the penalty function has been computed for each combination, the combinations are sorted by the value of their penalty functions. The algorithm then chooses the best combinations to place first.

The principal advantage of this method over the FAST algorithm is that it usually results in both a larger number of objects allocated as well as a higher sum of weights. For this reason, this is the preferred option for most fields. The disadvantage of this method can be its speed. For a reasonably well dispersed field (where each object may be reachable by 5 or 6 fibres) the algorithm takes about a minute on an unloaded Sun Classic workstation. This can get prohibitively long for fields with large numbers of objects which are centrally concentrated (say a globular cluster) where the number of fibres which can reach each object without breaking the placement rules gets to be very large (say 20 or so).

6 Getting hard copies using af2_hcopy

If you have a configuration and you would like to obtain a hard copy of it, you could get into **af2_configure** by using this configuration as an old configuration file and then creating the hard copy in the editor. If you don't want to go through all that, then **af2_hcopy** is a convenient shortcut. This program simply reads the old configuration file, the fibre information file and telescope information file and creates a hard copy plot file of the configuration as you would see it in the **af2_configure** editor. The questions it asks are all included in Section 3 in the list of prompts. The only thing to watch out for here is that you make sure that the PGPLOT device you specify is in fact a hardcopy device. If you specify a soft device, then an error will be signaled and the program will exit.

7 Installing af2_configure

The **af2_configure** package can be obtained either from the author or by anonymous ftp to the RGO (ftp.ast.cam.ac.uk). In the latter case the tar file can be found in the directory /pub/jrl/config, along with this document. The tar file contains all of the source code to **af2_configure** and recent versions of the fibre and telescope information files. **af2_configure** needs three main libraries before it can be linked. The first is Pat Wallace's astrometric library, slalib. If you are installing this at a STARLINK site, then you probably already have this available. If not, then a recent version of the library is provided in the **af2_configure** distribution. The same goes for the second library, which is the STARLINK 'chr' library. This is a suite of character handling routines of which **af2_configure** makes a lot of use. The third pre-requisite is PGPLOT which needs to be at either version 5.0 or above. There is a version of PGPLOT available at all STARLINK sites. Even so, the preferred option is to use the standalone version of PGPLOT which will plot *much* more quickly. This is available from Tim Pearson at Caltech through anonymous ftp (astro.caltech.edu or 131.215.240.1).

At present **af2_configure** has been tested on Suns running Sun/OS v4.1.3 or Solaris v5.4. It has also been tested on an ALPHA workstation running OSF1 v3.2.

Assuming that the **af2_configure** tar file has been brought over to your home institution it can be installed in the following manner:

1. Make a directory somewhere to hold the **af2_configure** directory tree and put the compressed tar file into it. Then issue the command:

   ```
   % zcat af2_configure.tar.Z | tar xpvf -
   ```

2. Type `% cd src` and get a listing of the directory. You should see a file called 'build'. Typing `% build` should cause the package to begin building.

3. The first thing that 'build' attempts to do is to locate a version of PGPLOT by looking at your environment variable `LD_LIBRARY_PATH`. If you don't have `LD _LIBRARY_PATH` defined, then it will search a couple of reasonably obvious locations (**/usr/local/lib** and **/usr/lib**). If it finds the PGPLOT library then it asks you if you want to use this version. If you do, then it proceeds to the next step. If not, it will search for a version in the STARLINK area. If it can neither use a standalone version nor find a STARLINK version, the build script bombs out.

4. The next question 'build' asks is whether or not you want to use the internal versions of 'chr' and 'slalib'. As we noted previously, the **af2_configure** package contains a recent release of the 'chr' and 'slalib' routines. If you answer 'yes' then these two internal libraries will be built. If you answer 'no' then it will search for these libraries in the STARLINK library area.

5. Finally, the **af2_configure** programs and scripts are built and installed. You should find three startup scripts, three binaries and some data files in the binary directory.

Once **af2_configure** has been built it can be accessed by putting the full path to the binary directory into the user's path (defined in the users .tcshrc file). Alternatively you could create soft links to the binaries and the scripts into a reasonably standard locations, *e.g.* **/usr/local/bin**.

A Common errors

I have listed here some of the most common errors which you may encounter when running **af2_configure**. They are listed below by the message they generate. There are other error messages which you might encounter. Ones with the words 'programming error' at the end probably mean just that and need to be referred to the author.

Error opening file: <filename> The file which you have tried to open doesn't appear to exist in the current directory. Are you sure the file has the correct extension (.fld for a field file)? This error may also occur when trying to open a new file. In that case, are you sure you haven't run out of disc space?

The file <file> does not exist. Do you have the correct bundle in field file? The fibre information file name is formed by appending the fibre bundle name to the string '_fibres.dat'. Thus if you don't have the correct fibre bundle name in the header of your field file, then the file that **af2_configure** tries to open will either be wrong or won't exist. The other possibility is that in the definition of WHT_AF2DATA_DIR you have given a directory where the fibre information file you want doesn't exist.

Requested PGPLOT device is not a soft device! This may occur if you specify a hard device (*e.g.* a postscript file) when the PGPLOT device is first requested for edit mode. If you want a hard copy, you should first display the configuration on a soft device and use the **h** command in the edit mode to create a hard copy. You could also use **af2_hcopy** if all you want is a hard copy.

Input field file has no object data! Are you sure there is any object information in your field file after the header?

Maximum version number exceeded for <filename> There is a maximum version number for configuration files, log files and x,y files. This is preset in **af2_configure** but may be changed by whoever builds the package. If you get this error message, try deleting the old versions of the offending file and try again.

Error reading line <lineno.> of <filename> An error has occured in reading from a file which is supposed to be in a set format. Check the offending line in the file. This is not always regarded as a fatal error.

Too many fibres specified in <filename> The maximum number of fibres which **af2_configure** will accommodate is preset, but can be changed by whoever builds the package. If you get this error, then check the number of fibres as specified in your fibre information file.

Error reading <headeritem> in <filename> The header items must be specified in the correct format (*e.g.*UTDATE must be a number and not a character string). Check the offending header item in the specified file and see if it makes sense.

<headeritem> is multiply defined in file <filename> A header item must only appear once in a file.

<headeritem> is undefined in file <filename> All header items must be defined. See if the offending header item exists in the specified file. Also check that it is written *before* any object information.

<headeritem> in old config file doesn't match that of field file If you are using an old configuration file as the starting point for your current run, you must have the same values for *all* your header items in both the old configuration file and the current field file. This is to ensure that a valid configuration which exists in an old configuration file isn't made invalid by different header values in the current field file.

Collision detected between fibres <fibno.> and <fibno.> This is a serious problem that may occur when reading an old configuration file. Presumably the old configuration file has a legal configuration, so if we detect a collision when reading it back in, it may indicate that someone has edited the file. It may also mean that the original configuration was done with a smaller buffer size than you have entered in your current run.

B Some programming notes

The current version of **af2_configure** is a very thorough revision/rewrite of a program of the same name by T.N. Wilkins at Durham (which was in turn based on a program written by I.R. Parry). I wish to stress that although almost all of the source code in this version is attributed solely to the author (done in most cases for simplicity), some of it is in fact based on the work of others, particularly in the design of the graphical routines.

This version will only work on UNIX-type platforms. There are several locations within **af2_configure** and its subroutines where searches are done for directories and these are only sought in a way which is consistent with UNIX path names. This could change in the future if there is sufficient demand.

Most of **af2_configure** is written in FORTRAN. In doing so, the author has tried to stay within the bounds of ANSI FORTRAN 77, but inevitably a few non-standard features have crept in. The main offenders are:

include: Include files are used extensively in **af2_configure** in order to define global parameters.

implicit none: This was used in order to make sure that all variables were declared.

do while: Used for the usual reasons.

%val: Used to pass pointers to workspace in subroutine calls.

long variable names: ANSI FORTRAN 77 specifies a maximum of 6 characters for the length of a variable name. This rule is broken fairly comprehensively in **af2_configure**.

$ format descriptor: This is used to suppress carriage control in the routine which prompts for values from the user (getval.f).

A small subset of routines have been written in ANSI C. These all have the prefix 'csub_' to their names and are used for operations which FORTRAN 77 can't do (*e.g.* allocating dynamic memory).

For the most part **af2_configure** is independent of the telescope and should be reasonably easily changed if a person wants to use it for a different fibre positioner or telescope. The following may need to be changed if **af2_configure** is to applied to another system:

af2_par.inc, af2_par.h These are include files for the FORTRAN and C routines respectively. If one is changed you must be careful to change the other as well. The parameters in these files define the maximum number of objects, the maximum number of fibres and many other purely internal parameters, which may never need changing.

af2_obs.inc This is an include file which is used by the astrometry routines. The parameters define various aspects of the observatory and telescope. The quantities are all documented in the file comments.

af2_fsta.inc This is an include file which defines the values of the fibre status variable (*e.g.* whether the fibre is good, broken, object, fiducial or whatever) for each type of fibre. **af2_configure** inherited these definitions from the autofib-2 micro software and the fibre information file has the fibre status defined in terms of these values.

platexy.f, centrexy.f, xytord.f These are the three main astrometric routines. For the most part these should be the same for all telescopes. What may need changing is the definition of the rotator angle and the direction cardinal axes of the instrument. These sections are commented in the code.

Information on object and fibre positions is carried around in two types of data what I call here 'static' and 'variable'. The static data have information which never changes after they have been read in from the input files. The variable objects are changed and rearranged during the course of the fibre placement. These data are called almost without exception by the same variable names in all routines in **af2_configure**. A list of the main static information regarding the objects is given below:

nobj, nsky These are the number of objects and the number of sky patches which were read in from the field file. Note that all fiducial objects are included in **nobj** and that both will exclude objects/skies which fall outside the field. Integers.

object, skypos These are arrays of the x,y coordinates of the objects and sky patches respectively. They will have been derived from the (α, δ) included in the input field file. Real arrays

otype This is an object type for the **object** array and is used to differentiate between program and fiducial objects. The values are set to either 'fsta_ook' or 'fsta_fok', as defined in the file 'af2_fsta.inc'. Integer array.

plog, slog These are arrays which carry around almost the whole of the input field file. Each element is an 80 character string which contains the input line from the field file for a particular object (program or fiducial) or sky patch. I quite often refer to these as the 'object information log' in the comments. Character arrays.

weight, sweight These are the arrays which carry around the defined weights for the objects and sky patches respectively. Real arrays.

The important static information regarding the fibres is:

fibstatus This is an array of values which describe the fibre's status. By 'status' we mean its availability for use, *e.g.* if the fibre is broken then it is flagged as such. The values it can have are described in the include file 'af2_fsta.inc'.

piv This is the array of x,y positions for each of the fibre pivots. Integer array.

park This is the array of park positions for each of the fibres. Integer array.

The variable information is what essentially describes the configuration. This is covered in the list below:

object_ref, sky_ref These are referred to repeatedly in the code as the object reference list and the sky reference list. They contain a list of objects/skies which have not yet been covered by a fibre in the current configuration. For various reasons these arrays may be rearranged and it is important to realise that the value in a reference list refers to a particular object/sky in the static information. For example, if the object reference list has been sorted in such a way that the first item in the list is the one which would best suit a particular fibre. Then if **object_ref(1)** equals 30, this means that the object whose x,y coordinates are in **object(1,30)** and **object(2,30)** and whose weight is stored in **weight(30)**, and so on is the one that is best suited for the fibre in question. Integer arrays.

n_obj_left, n_sky_left These are the number of object and sky patches respectively which have not yet been covered by a fibre. Another way of looking at this is that they are the number of objects still available in the object and sky reference lists. Integers

fib This is the current x,y coordinates of each of the fibres (in microns from the centre of the field). Integer arrays.

allocated Initially all of the elements of this array are give a value of zero. This is to indicate that the fibre is available to be used for an object. If the fibre is disabled, dead or missing, then the element of this array corresponding to that fibre is given a value of -10, -20 or -30 respectively. Any fibre with a negative value in this array is automatically ignored for configuration and editing purposes. If the fibre is later allocated to a particular object, then the value of this array for that fibre is changed to the reference list number of the object in question. If the fibre is allocated to a sky patch then the appropriate element of this array is set to the reference list number of the sky patch, plus a standard offset value which is called **alloc_sky** internally. If the fibre is allocated to a cursor location in the editor, then the value of this array is set at a single standard value, called **alloc_cur**. Both of these standard values are set in the include file 'af2_par.inc'. It is quite often the case that a routine in this package will use the value of this array to determine what type of object is being covered. Take, for example a value of *ix* for a particular element of this array. If *ix* equals **alloc_cur** then we know the fibre was placed on a cursor location in the editor. If its value is greater than **alloc_sky**, then we know its placed on the sky patch whose reference number is *ix* - **alloc_sky**. If *ix* is greater than zero and less than **alloc_sky** then *ix* is the object reference number for a program or fiducial object. Integer array.

All of the arrays in the three lists above have been dimensioned to some maximum number of elements. The actual dimensions are listed in the include file 'af2_par.inc'. I realise that this is inelegant in the extreme and that I should have written it so that the space was allocated dynamically at run time (in the same way that the workspace is allocated for the placement routines). But this was the way that the original program was written and to make such a fundamental change would have been a major piece of surgery which time didn't allow.

C Acknowledgments

The author wishes to thank Paul Rees, Brian Boyle, Terry Bridges, Fred Watson, Ian Parry, Ian Lewis and Don Pollaco for a lot of good advice and encouragement. Thanks also to Ralph Martin and Bob Thomson for agreeing to be guinea pigs with the beta release of this software and for offering many suggestions for improvement. Finally a big 'thank you' to Carole and Jessica for putting up with the author's obsession with this project over the past 3 months.