

Making and Editing File Lists from the Command Line

IRAF works very well at processing batches of files from a simple text list. Manually creating a file list by typing in the files individually can be tedious, so here are some tips on using command line tools and text editors to edit the lists more expediently.

In Linux, Mac terminal, or Windows Xming, WSL (or similar) command line prompt you can use simple commands to create a list of files, e.g.

```
ls *.fits > allfiles.list
```

ls is a command that does a directory listing of the following matching files.

*.fits is a wildcard that matches any file ending in .fits. Other wildcards are ? which is a single character wildcard. Similarly, one can use square brackets to include a list of characters or range of characters to match. For example d10[0-3].fits will match d100.fits, d101.fits, d102.fits, and d103.fits. d10[14a].fits will match d101.fits, d104.fits, and d10a.fits.

> allfiles.list takes the output of the command before it and puts it into a plain text file instead of printing it to the terminal output.

To look at the contents of the newly created file, you can use cat, more, or less, e.g.

```
cat allfiles.list
```

The resulting output will look something like the following:

```
b100.fits  
b101.fits  
b102.fits  
b103.fits  
b104.fits
```

Many of the IRAF scripts require input and output file name lists so it knows the original file name and the new file name for the processed image (IRAF does not overwrite file names by default).

In unix/linux there are powerful command line programs like sed and awk that allow one to do a global search and replace in a text file and pipe the output to a new file, to easily create a list of new output filenames from an initial file listing file names.

Using sed, the command for global search for the text string .fits and replacing it with _os.fits in the file allfiles.list, and writing it out to the new file allfiles_os.list would be:

```
sed 's/.fits/_os.fits/g' allfiles.list > allfiles_os.list
```

The command `awk` can do the same thing with the following command:

```
awk '{gsub(/.fits/, "_os.fits"); print}' allfiles.list > allfiles_os.list
```

It is sometimes easiest to copy the input file name list to a new file and then edit that new file using a text editor, often `vim` (or `vi`) or `xemacs` (or `emacs`). To copy a file at the command line, use the `cp` command, e.g.

```
cp allfiles.list allfiles_os.list
```

There are now two identical files named `allfiles.list` and `allfiles_os.list`. You will want to edit the second file, `allfiles_os.list` to change the output file names to something meaningful and distinct from other file names. First open the file in your preferred text editor. In this first example, we'll use `vim`.

```
vim allfiles_os.list
```

It is most expedient in this case to do a global find and replace. It is easy to replace `.f` in each file name with `os.f` (you can use anything you want, but in keeping with the companion IRAF tutorial document this example is making a list of raw data files and creating overscan subtracted files, so using `os` as shorthand for overscan subtracted is useful to denote the content of the file).

In `vim` or `vi` to do a global substitution you use the `:s` command, e.g.

```
:%s/.f/os.f/g
```

The `%` tells it to search and replace every occurrence of the pattern in the file.

The `/` separates the string to look for from what it is to be replaced with.

The `g` tells it to replace the string every time it occurs in a single line.

Other useful commands in `vim` (or `vi`) are:

`i` Enter insert mode, which allows you to move the cursor around and type in or delete text where you want to put it. `Esc` gets you out of insert mode.

`dd` Deletes the line the cursor is on.

`x` Deletes the character the cursor is on.

`u` Undoes the previous action.

`w` Writes the file

q Quits vim or vi

Another available editor is emacs or xemacs. xemacs is a more graphical user interface (GUI), which may be more comfortable for some users.

xemacs allfiles_os.list

This will bring up a GUI with various icons for saving the file, cutting or pasting text, etc. However, the keyboard commands are very handy for global search and replace to be used in this case.

Make sure your cursor is at the first line and character in the file, then type the following to get into the command buffer (the line at the bottom of GUI)

Esc-x

Then type

replace string

This will prompt you for the string to be replaced, so in this case type

.f<return>

You will then be prompted for the string to replace with, so in this case type

os.f<return>

This will now replace every .f in the file with os.f

You can now save the file either by using the GUI save icon or typing Ctrl-x Ctrl-s (or save and quit with Ctrl-x Ctrl-c).