A basic AIPS help sheet
=======================
This is intended as a useful memory aid, not a user guide. If you are new to AIPS and want to do any serious imaging then it is highly recommended that you read through the relevant sections of the cookbook, available as hardcopy in any of the student offices, or online at the NRAO's website (www.nrao/edu/aips/).

Before starting aips
====================
```
    setenv DATA '/some/dir'    - set the environment variable DATA to
                                      point to /some/dir
```
Starting up
===========
```
    aips                       - start up aips on your machine
    aips notv                  - start aips without a tv
    aips tv=local:0            - start aips in a vnc session
```

Aips tasks, verbs, etc
======================
Aips has a huge collection of commands. Some are called tasks, but there are also verbs, adverbs and procedures. Tasks are run in the background allowing you to continue typing other commands in aips (unless you set DOWAIT=1). Verbs however are generally shorter programs which run in the foreground. Procedures are like scripts that run a series of commands. Adverbs set parameters that tasks, verbs and procedures use. Note that the names of aips commands can be shortened as much as you like, as long as the sequence is still unique. For example, tvinit can be shortened to tvin, but not to tv because there are other tasks such as tvstat and tvwin. To run tasks, verbs and procedures, type
```
    > go TASK
    > VERB
    > run PROCEDURE
```
When you start aips, another three windows open: a tv, a message server and a tek server. To make sure a command sends its output to the tv (rather than to a plot file) make sure you set
```
    > dotv 1
```
or, for text based tasks, to make sure the output goes to the command window (rather than the printer or a file), set
```
    > docrt 1
```
You can also talk to aips (and use it as a calculator) using POPS with commands such as
```
    > for i=1 to 10; getn i; zap; end
```
which will delete files 1 to 10 in your catalogue (type "about pops" for more info).

Getting help
============
```
    > inp TASK       - show the current inputs for TASK
    > help TASK      - get help with the task TASK (or verb, adverb, etc)
    > explain TASK   - get help with the task TASK (or verb, etc...)
```

To check the inputs
===================
```
    > inp TASK       - check the inputs for a task (always a good idea)
    > tget TASK      - get the parameters that were used last time the
                        task was run or set with tput
    > tput TASK      - save the parameters currently set for task
    > restore 0      - reset all inputs to their default values
    > save THISSTATE - save the inputs currently set for all tasks
    > get THISSTATE  - retrieve the state saved as THISSTATE
    > sgindex        - lists all saved states
```

Files
=====
```
    > infil'DATA:file';go fitld   - read in a fits file from disk
    > outfil'DATA:file';go fittp  - write a fits file to disk
    > pcat                        - list all the files in your catalogue
    > getn n                      - select the file specified by n
    > imhead                      - look at the fits header
    > prthi                       - print the file's history file
    > go prtan                    - print the antenna table
    > zap                         - delete the specified file (NO UNDO!)
    > extdest                     - delete extension of a particular type
                                     and version number (check with imh)
```
Simple display commands
=======================
```
    > tvlod                - load the selected image onto the tv display
    > tvinit               - clear all graphics and images from the tv
    > tvwin                - draw a window on the tv
    > tvfiddle             - adjust the grey scale and zoom on the tv
    > tvpseudo/tvphlame    - adjust the colour of the image on the tv
    > grcl                 - clear any graphics on the tv
    > imstat               - report statistics from a window on the tv
    > tvstat               - report statistics from a region selected by
                              the cursor
    > curval               - report pixel location and value at the
                              current cursor position
    > tvlabel              - put axes labels (RA, Dec) on the tv
```

Displaying uv data
==================
```
    > go listr                  - display a table in the command window
    > optype 'scan'; go listr   - displays a scan listing
    > optype 'matx'; go listr   - prints the visibilities as a matrix
    > go possm                  - displays cross or total power spectra
    > go uvplt                  - plot visibilities on the tv
    > go vplot                  - plot uv data per baseline
    > go snplt                  - plots SN/TY/CL tables
```

```
Flagging commands
=================
    > about editing      - show a list of useful editing commands
    > go edita           - interactive editing based on TY/SN/CL tables
    > go tvflg           - interactive flagging using the tv
    > go spflg           - interactive baseline based spectral editor
    > go ibled           - interactive baseline based visibility editor
    > go snedt           - interactive SN/CL table editor using the TV
    > go uvflg           - flag uv data based on inputs (non-interactive)
    > go wiper           - interactive flagging in a uvplt-like display

Some housekeeping commands
==========================
    > go msort/uvsort    - sorts data e.g. in to time-baseline order
    > go uvfix           - recomputes uv coordinates
    > go split           - create single-source files (applying
                           calibration if requested)
    > go splat/uvcop     - copy uv data averaging or applying calibration
    > go dbcon           - combine two uv databases into one
    > go uvhgm           - show statistics of uv data

Some calibration commands
=========================
    > about calibrat     - show a list of useful calibration commands
    > go setjy           - set the flux of the main calibrator(s)
    > go getjy           - determine other calibrator flux densities
    > go calib           - determine complex gains
    > go clcal           - merges SN tables to create a CL table
    > go fring           - fringe fit UV data
    > go bpass           - bandpass calibration

Some imaging commands
=====================
    > about imaging      - show a list of useful imaging commands
    > go imagr           - FTs uv data and (optionally) cleans the image
    > go apcln           - deconvolves dirty map/beam
    > go vtess           - deconvolves using a MEM method
    > go regrd           - regrids an image to another coordinate frame

Some spectral commands
======================
    > about spectral     - show a list of useful spectral commands
    > go ispec           - plot the spectrum of the area boxed by tvwin
    > go tvmov/tvcub     - step through the cube on the tv like a movie
    > go trans           - transpose a cube
    > go uvlin/uvlsf     - remove the continuum from uv data cube
    > go imlin           - remove continuum from an image cube
    > altdef/altsw       - add/switch between velocity and frequency in
                           the FITS header
    > go slice           - generate a spectrum for part of a cube
    > go tksl            - display the SL file generated by slice

Some plotting commands
======================
    > about plot         - show a list of useful plotting commands
    > go uvplt           - plots uv data
    > go cntr            - make a contour plot
    > go ccntr           - like cntr, but also plots ST, CC or MF cmpts
    > go kntr            - make a contour/grey scale plot
    > go tkpl            - send a plot file to the TEK server
    > go tvpl            - send a plot file to the TV

Some fitting commands
=====================
    > imfit/jmfit        - fit a section of an image using up to four
                           Gaussian components
Output
======
    > dotv -1; go TASK             - make a plotfile using the output
                                     from TASK
    > outfil'file.ps'; go lwpla    - send the last plotfile to a file
                                     specified by /tmp/file.ps
    > outfil''; go lwpla           - send the last plotfile to the
                                     printer
    > outprint'file'; prtmsg       - print the message buffer to the
                                     file specified by 'file'
    > clrmsg                       - clear the message buffer (aips will
                                     complain when it gets too full)
Backup
======
    > outfil'file'; go fittp       - write a fits file to disk
    > tasav                        - save the tables of a uv dataset

To quit
=======
    > exit                         - exit aips
    > kleenex                      - quit aips and kill all windows


ParselTongue
============
Requires a manual all of its own.  But, to run the e-MERLIN pipeline
(which is just a ParselTongue script) from the command line you need to
type the following:
    $ ParselTongue eMERLIN_pipeline.py inputs.txt
where inputs.txt is a plain text file containing the options required to
drive the pipeline.  See http://www.e-merlin.ac.uk/observe/pipeline/



20040708 mkargo
Last updated: 20140115
```