

# Utility programs and scripts

Thomas Herring

[tah@mit.edu](mailto:tah@mit.edu)

# Utility Overview

- In this lecture we look at a number of utility scripts and programs used in the gamit/globk suite of programs.
- We examine and will show examples in the areas of
  - Organization/Pre-processing
  - Scripts used by sh\_gamit but useful stand-alone
  - Evaluating results
- Also examine some basic unix, csh, bash programs and method.

# Guide to scripts

- There are many scripts in the `~/gg/com` directory and you should with time look at all these scripts because they often contain useful guides as to how to do certain tasks.
  - Look the programs used in the scripts because these show you the sequences and inputs needed for different tasks
  - Scripting methods are useful when you want to automate tasks or allow easy re-generation of results.
  - Look for templates that show how different tasks can be accomplished.
- `~/gg/kf/utills` and `~/gg/gamit/utills` contain many programs for utility tasks and these should be looked at to see what is available.

# GAMIT/GLOBK Utilities

## 1. Organization/Pre-processing

sh\_get\_times: List start/stop times for all RINEX files

sh\_upd\_stnfo: Add entries to station.info from RINEX headers

convertc: Transform coordinates (cartesian/geodetic/spherical)

glist: List sites for h-files in gdl; check coordinates, models

corcom: Rotate an apr file to a different plate frame

unify\_apr: Set equal velocities/coordinates for *glorg* equates

sh\_dos2unix: Remove the extra CR from each line of a file

doy: Convert to/from DOY, YYYYMMDD, JD, MJD, GPSW

## GAMIT/GLOBK Utilities (cont)

### 2. Scripts used by sh\_gamit but useful stand-alone

sh\_get\_rinex: ftp a RINEX o file from remote archives ( ftp\_info)

sh\_crx2rnx: convert to/from RINEX/Hatanaka

sh\_get\_nav: ftp a RINEX n file from remote archives

sh\_get\_met: ftp a RINEX m file from remote archives

sh\_get\_hfiles: ftp h-files from SOPAC/MIT

sh\_update\_eop: ftp an EOP file from IERS, create pmu, ut1., wob.

sh\_get\_orbits: ftp a g-file or sp3 file from remote archives, call -->

sh\_sp3fit: create a g- or t-file from an sp3 file (1-3 days)

## GAMIT/GLOBK Utilities (cont)

### 3. Evaluating results

- sh\_oneway: Plot phase residuals (sky map; vs elevation) [ GMT]
- sh\_plotelmean: Elevation angles average residuals plots [ GMT ]
- cview: Display and manipulate phase residuals [X-windows]
- sh\_plotcrd: Plot coordinate times series [GMT]
- sh\_tshist: Plot histogram of time-series statistics [GMT]
- tsview: Display and manipulate coordinate time series [MATLAB]
- sh\_plotvel: Plot velocity maps [GMT], call -->
- sh\_map\_elements, sh\_map\_calif, sh\_map\_balkans, etc.
- velview: Display and manipulate velocity maps [MATLAB]
- sh\_org2vel: Extract plate-referenced velocities from glorg print file
- velrot: Combine velocity fields from different analyses

# Unix Primer

- There are web resources with primer's on the Unix system and you should consult some of these resources. Often web-searching for a unix command will yield useful results (links in notes below).
- Unix command are executed in a shell. The main ones are csh (or tsch) and bash. GAMIT/GLOBK scripts are all csh scripts (scripts are shell commands put in files so that they can be executed many times: Automates tasks).
- Your setup for your shell is defined in either ~/.cshrc or ~/.bashrc (~ is symbol for home directory). Setting up these files is needed for gamit.
- Looking at these scripts can give you some idea of the commands

# .cshrc/.bashrc

## CSH commands

```
# CSH Gamit/Globk initialization
# Basic setup
set ggdir = $HOME/gamit_10.40
setenv HELP_DIR = ~/{ggdir}/help/
setenv GMTHOME = $HOME/GMT/GMT3.4.4
set INSTITUTE = MIT
set prompt = %m["%h"] "
set noclobber # Use >! Overwrite
set history=1500
set savehist=1500
```

## BASH commands

```
# BASH Gamit/Globk profile
# Basic setip
ggdir=$HOME/gamit_10.40
export HELP_DIR=${ggdir}/help/
GMTHOME=$HOME/GMT/GMT3.4.4
export GMTHOME
INSTITUTE=MIT ; export INSTITUTE
PS1="\h[\#] "
set -o noclobber # Use >| overwrite
export HISTFILESIZE=1500
```



# .cshrc/.bashrc

## Path setup .cshrc

```
set spath = ( /usr/local/bin /usr/local/sbin /usr/bin /bin \  
             /usr/sbin /sbin /usr/local/gfortran/bin /usr/X11R6/bin )  
set gpath = ( $ggdir/com $ggdir/gamit/bin $ggdir/kf/bin \  
             $ggdir/matlab/tsview $ggdir/matlab/velview $ggdir/python)  
set lpath = ( . /Users/tah/bin /opt/ibmcmp/xlf/8.1/bin)  
set path = ( $lpath $gpath $spath)  
set path = ( /sw/lib/gcc4.4/bin/ $path $GMTHOME/bin )
```

## Path setup .bashrc

```
spath=/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/X11R6/bin  
gpath=$ggdir/com:$ggdir/gamit/bin:$ggdir/kf/bin:\  
$ggdir/matlab/tsview:$ggdir/matlab/velview:$ggdir/python  
lpath=$HOME/bin:.  
upath=/sw/bin:/sw/lib/gcc4.4/bin:$GMTHOME/bin  
PATH=$spath:$gpath:$lpath:$upath ; export PATH
```

# .cshrc/.bashrc

## Aliases

```
# ALIASES
alias cp 'cp -i'
alias CP '\cp\'
alias mv 'mv -i'
alias rm 'rm -i'
alias dir 'ls -la'
alias ls 'ls -F'
alias RM '\rm\'
alias pd pushd
alias xt 'xterm -sb -sl 2000'
alias H 'history'
alias m more
alias ll 'ls -la'
alias ne 'nedit \!* &'
alias last 'ls -lFt \!* | head -20'
```

## Aliases

```
# ALIASES
alias cp='cp -i'
alias CP='\cp\'
alias mv='mv -i'
alias rm='rm -i'
alias dir='ls -la'
alias ls='ls -F'
alias RM='\rm\'
alias pd=pushd
alias xt='xterm -sb -sl 2000'
alias H='history'
alias m=more
alias ll='ls -la'
function ne() { nedit "$@" & }
alias last='ls -lFt $@ | head -20'
```

# Command Unix commands

- man to get help e.g., man ls
- Directory commands:
  - ls list directory (\* and ? Match any characters (\*) or any single character (?))
  - cd change directory
  - pwd print current (working) directory
  - mkdir make directory
  - rm remove file (\rm to not ask of rm aliased to rm -i (interactive remove))
  - rmdir remove directory (must be empty)

# Unix editors

- Text editors: Mac textedit is OK (save as text)
- Other common full screen editors:
  - gedit (most Linux systems)
  - nedit (one download source:  
[http://ftp.nl.freebsd.org/editors/NEdit/v5\\_5/executables/](http://ftp.nl.freebsd.org/editors/NEdit/v5_5/executables/))
  - vi or vim available on all platforms and is good to know.
- Line editor: Some commands (:help for help)
  - i for insert mode, esc to exit this mode
  - : for command mode (:w write, :q quit, :q! quit no save)
  - o insert line start input, A append to end of line.
  - Arrow keys to move around page
  - / to search
  - Start:end s /<str>/<rep>/g Form line start (1) to line end (\$ for last line), replace <str> with <rep>. Append g for multiple replacement.

# Unix csh program/commands

- `grep` – used to find content in files. We use it to extract information from files
- `awk` – used to extract column based data from files. `awk` has math, logical and string functions.
- Pipes and re-directs: These methods differ between `csh` and `bash`:
  - `csh`: `>`, `>>` and `>&`, `>>&` to re-direct stdout and stdout+stderr
  - `csh` `>!` will overwrite file when `noclobber` set
  - `bash`: `>`, `>>` and `>> file 2>&1` to do the same
  - `bash` `>|` will overwrite file when `noclobber` set
- `setenv` and `set` allow variables to be set and differ between `csh` and `bash` (`setenv` and `export` create global variables that can be seen in other)
  - `csh`: `set variable = value ; setenv variable value`
  - `bash`: `variable=value ; export variable`
  - In both case `$variable` contains the value

# Summary

- We have looked at just some examples of common scripts and program used in gamit/globk
- There are many more scripts to be found in ~/gg/com and programs in ~/gg/gamit/bin and ~/gg/kf/bin
- A good understanding of unix csh or tcsh is very useful. The software will run from a bash shell but all the instructions are for csh