An Introduction to IRAF and the Gemini IRAF package

Emma Hogan, Gemini Observatory

What is IRAF?

- **I**mage **R**eduction and **A**nalysis **F**acility
  - Provides a wide range of image processing tools using a command line interface
- Documents
  - `http://iraf.net/irafdocs/`
- Beginner’s guide
  - `http://iraf.net/irafdocs/beguide.pdf`
- IRAF support
  - `http://iraf.net/`

What is Gemini IRAF?

- The Gemini IRAF package is one of many external IRAF packages
- Objective: To provide the tools necessary to remove instrument and atmospheric signatures from the data for **all** Gemini facility instruments
- Information about the Gemini IRAF package
  - ➢ `http://www.gemini.edu/sciops/data-and-results/processing-software`
- Gemini IRAF support
  - ➢ `http://www.gemini.edu/sciops/helpdesk/`

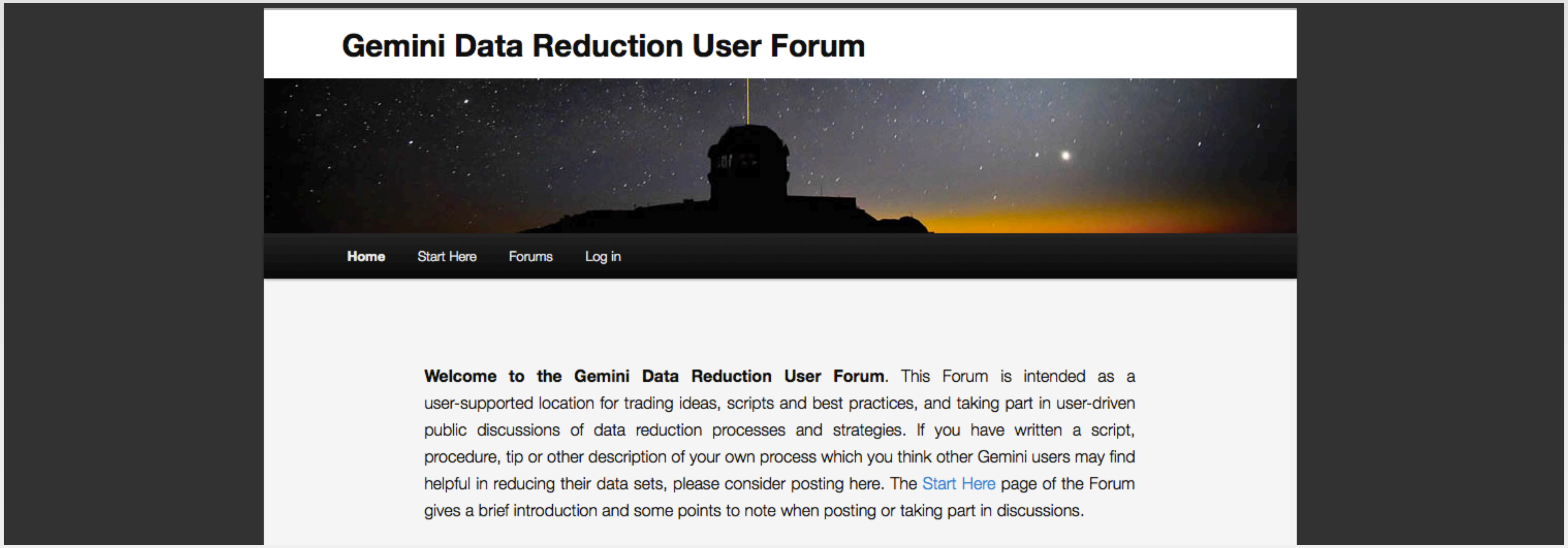
GEMINI OBSERVATORY
Exploring the Universe, Sharing its Wonders

Data Reduction Support

- The Getting Started web page
  - ➢ `http:/www.gemini.edu/sciops/data-and-results/getting-started`
- The Data Reduction Support web page include an FAQ and a list of known problems
  - ➢ `http://www.gemini.edu/sciops/data-and-results/data-reduction-support`
- Also check the Announcement web page
  - ➢ `http://www.gemini.edu/sciops/data-and-results/processing-software/announcements`

Data Reduction User Forum

- The Gemini Data Reduction User Forum was made publicly available at the end of 2013
  - ➢ `http://drforum.gemini.edu/`



Welcome to the Gemini Data Reduction User Forum. This Forum is intended as a user-supported location for trading ideas, scripts and best practices, and taking part in user-driven public discussions of data reduction processes and strategies. If you have written a script, procedure, tip or other description of your own process which you think other Gemini users may find helpful in reducing their data sets, please consider posting here. The Start Here page of the Forum gives a brief introduction and some points to note when posting or taking part in discussions.

Demo: Set up

- Open a terminal window:

```
% xterm
% xgterm          # use -sb for scrollbar
```

- Create a directory that will be the IRAF home directory:

```
% cd
% mkdir iraf
% cd iraf
```

- Any directory can be the IRAF home directory
  - ➢ but PyRAF automatically looks in `$HOME/iraf`

GEMINI OBSERVATORY
Exploring the Universe, Sharing its Wonders

Demo: Set up

- Create the IRAF startup files (only need to do this once)

```
% mkiraf
```

- What does `mkiraf` do?
  - ➢ Creates a `login.cl` file and `uparm` directory (this is where the user parameters are saved)
  - ➢ At the prompt, choose whether to initialize the `uparm` directory (if asked) and select the terminal type you are using
- What happens if I "Initialize uparm"?
  - ➢ The saved user parameters located in the uparm directory are deleted

GEMINI OBSERVATORY
Exploring the Universe, Sharing its Wonders

Demo: Starting IRAF

- Start an image display server
```
    % ds9 &
    % ximtool &
```
- Start IRAF from the IRAF home directory (the login.cl file is executed at this point)
```
    % cl
```
- Find out what core IRAF packages are available
```
    cl> help
```
- Find out more information about a particular package
```
    cl> help images
    cl> help gemini
```

GEMINI OBSERVATORY
Exploring the Universe, Sharing its Wonders

Demo: Gemini IRAF

INSTRUMENT	MODES	PACKAGES
FLAMINGOS-2	imaging longslit	`f2` / `gnirs`
GMOS	imaging longslit MOS IFU	`gmos`
GNIRS	longslit XD IFU	`gnirs`
GSAOI	imaging	`gsaoi`
NIFS	IFU	`nifs` / `gnirs`
NIRI	imaging longslit	`niri` / `gnirs`

Demo: Packages

- To load a package, just type the package name

```
    cl> gemini
```

- Loading a new package does not unload the previous package

```
    cl> gmos
```

- Unload the last package that was loaded

```
    cl> bye
```

- Find out what packages are loaded

```
    cl> package
```

- The prompt reflects the last package loaded

GEMINI OBSERVATORY
Exploring the Universe, Sharing its Wonders

Demo: Tasks

- Find out what tasks are in the currently loaded package

```
    cl> ?
```

- Find out what tasks are currently loaded

```
    cl> ??
```

- Find out what tasks / packages are in a currently loaded package

```
    cl> gmos
    cl> ?gmos
```

- Find out more information about a particular task

```
    cl> help gprepare
```

- The help pages show which package a task belongs to

Demo: Parameters

- Most tasks have parameters that the user can modify to affect the output of the task
- List the parameters for a task

```
cl> lpar gprepare
```

- Required parameters must be specified each time the task is executed (shown without parentheses)

```
inimages=""  Input GMOS images or list
```

- Hidden parameters are shown inside parentheses and have a default value

```
(outpref="g")  Prefix for output images
```

Demo: Parameters

- Edit the parameters for a task

```
cl> show editor
cl> epar gprepare
```

- Edit a parameter value (using vi/emacs as default editor)

```
<ctrl>-u <ctrl>-l / <esc>-<ctrl>-k
```

- Exit epar and discard any changes

```
:q!
```

- Exit epar and save the changes to the uparm directory

```
:q
```

Demo: Parameters

- Set a parameter value on the cl command line

```
    cl> gprepare.outpref = "hello"
```

- This updates the parameter value for the session (but the parameter file in the uparm directory is **not** updated)

```
    cl> lpar gprepare
```

- Restore the original default parameters for a task

```
    cl> unlearn gprepare
```

- `unlearn` also deletes the appropriate parameter file from the uparm directory

GEMINI OBSERVATORY
Exploring the Universe, Sharing its Wonders

Demo: Executing Tasks

- When a task is executed, IRAF first searches the uparm directory for a customized parameter file. If one does not exist the system default file is used
- Execute a task from epar with the current parameters

```
:go
```

- Execute a task from the cl command line

```
cl> gprepare
```

- If a required parameter is not defined, a prompt will appear

```
Input GMOS images or list:
```

Demo: Executing Tasks

- Required parameters must appear on the cl command line in the order that they appear in the parameter list

```
    cl> cd /path/gmos_image_tutorial
    cl> gprepare S20030525S0164.fits
```

- It is not necessary to specify the parameter name for required parameters
- Hidden parameters can appear in any order after the required parameters (since they include the name of the parameter)

```
    cl> gprepare S20030525S0164.fits /
            outimages=myoutput.fits
```

GEMINI OBSERVATORY
Exploring the Universe, Sharing its Wonders

Demo: Executing Tasks

- Required parameters specified on the cl command line are stored in the parameter files in the uparm directory
- This is not true for hidden parameters; the command line values simply override the defaults for that execution of the task

```
    cl> lpar gprepare
```

- If any hidden parameters are not specified at execution time the current parameter values will be used

GEMINI OBSERVATORY
Exploring the Universe, Sharing its Wonders

Demo: Aborting Tasks

- Abort a task

```
    <ctrl>-c
```

- Sometimes things can be left in a weird state after an abort
- It is generally good practice to execute flprcache a few times after an abort

```
    cl> flpr
```

- If problems still occur, log out of IRAF and then back in

Demo: History

- Access history

  use the up arrow

```
cl> e gprepare
cl> history
```

- Save all commands from the current IRAF session

```
cl> history -999 > my_history.txt
```

- Execute the previous command

```
cl> ^^
```

- Execute a particular command number

```
cl> ^35
```

Demo: MEF files

- Gemini data are in the form of **M**ulti-**E**xtension **F**ITS files
- The IRAF tasks expect single extension FITS files
    - ➢ header and pixel data contained in a single extension
- The tasks in the Gemini IRAF package expect MEF files
    - ➢ the first [0] extension is the **P**rimary **H**eader **U**nit (PHU) and contains keywords applicable to the pixel data as a whole
    - ➢ additional extensions contain pixel data and some header information specific to that pixel data

GEMINI OBSERVATORY
Exploring the Universe, Sharing its Wonders

Demo: MEF files

- The `fitsutil` external package contains tasks that can manipulate MEF files

```
    cl> help fitsutil
```

- Use `fxhead` to list a one line header description for each extension in the MEF

```
    cl > fxhead S20030525S0164.fits
```

- To work with Gemini data using IRAF directly, the index / extension number must be specified

```
    cl> imhead S20030525S0164.fits[0]
    cl> imstat S20030525S0164.fits[1]
```

GEMINI OBSERVATORY
Exploring the Universe, Sharing its Wonders

Demo: `gemtools`

- The `gemtools` package contains lots of cool stuff!
  - ➢ `gemarith` (`imarith`)
    - arithmetic on MEF files
  - ➢ `gemexpr` (`imexpr`)
    - image expression evaluator, handles MEF files
  - ➢ `gemcombine` (`imcombine`)
    - combine MEF files by extension
  - ➢ `wmef`
    - convert single extension fits files to MEF files
  - ➢ `gemlist`
    - generate a list of file names in standard Gemini format

Demo: Displaying images

- Display example image

```
    cl> display dev$pix 1
```

GEMINI OBSERVATORY
Exploring the Universe, Sharing its Wonders

Demo: Graphics

- Examine the example image

```
    cl> imexam dev$pix
```

- List the cursor options with `?`
- Try `a`, `l`, `r`, `s`
- Change to the graphics cursor with `g`
- Zoom in with `z`
- Return to the original plot with `0` (zero)
- Change to the image cursor with `i`
- Quit (return the the cl command line) with `q`

GEMINI OBSERVATORY
Exploring the Universe, Sharing its Wonders

Demo: Aborting Tasks ... Again

- The clean way to quit from a graphics task is to press `q` in the irafterm window
- If `<ctrl>-c` (or some other way) was used to abort the graphics task, things **will** be left in a weird state
- In this case, it is necessary to log out of IRAF, close and restart the image display server, and then log back into IRAF

Demo: Gemini IRAF

- Find out how to reduce your data

```
cl> gmosinfo
cl> gmosinfospec
```

- Want to follow an example reduction script?

```
cl> gsaoiexamples imaging
cl> gnirs
cl> gnirsexamples
```

Demo: Preparing Data

- All raw data must be prepared
  - ➢ Raw data is validated
  - ➢ Keywords are added to the PHU, e.g., `NSCIEXT`
  - ➢ Keywords are added to / corrected in the pixel data extensions, e.g., `GAIN`
  - ➢ Non-linear pixels are corrected, if requested
  - ➢ Variance and data quality extensions are added, if requested
  - ➢ An MDF will be added, if appropriate

Demo: Gemini Data

- Once prepared, extensions are named and versioned:

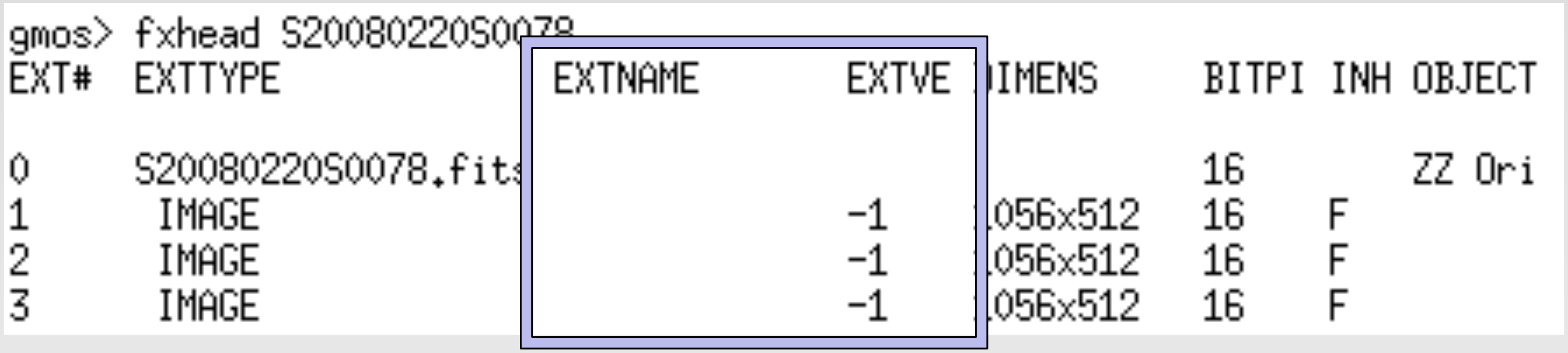
➢ `EXTNAME`	(extension name)
➢ `EXTVER`	(extension number)

- The value of the `EXTNAME` keyword could be:

➢ `SCI`	(Science)
➢ `VAR`	(Variance)
➢ `DQ`	(Data Quality)
➢ `MDF`	(Mask Definition File)

Demo: Gemini Data

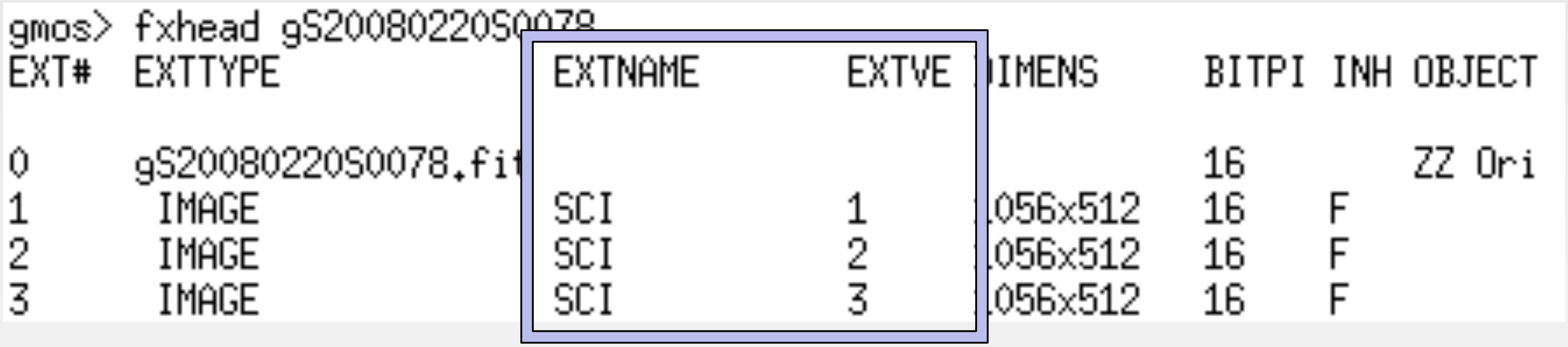
- Use the task `fitsutil.fxhead` to see the structure:
  - For raw GMOS data:



```
gmos> fxhead S20080220S0078
EXT#  EXTTYPE          EXTNAME        EXTVE  IMENS      BITPI  INH OBJECT

0     S20080220S0078.fit                                16         ZZ Ori
1      IMAGE                          -1     056x512    16     F
2      IMAGE                          -1     056x512    16     F
3      IMAGE                          -1     056x512    16     F
```

  - For prepared GMOS data:

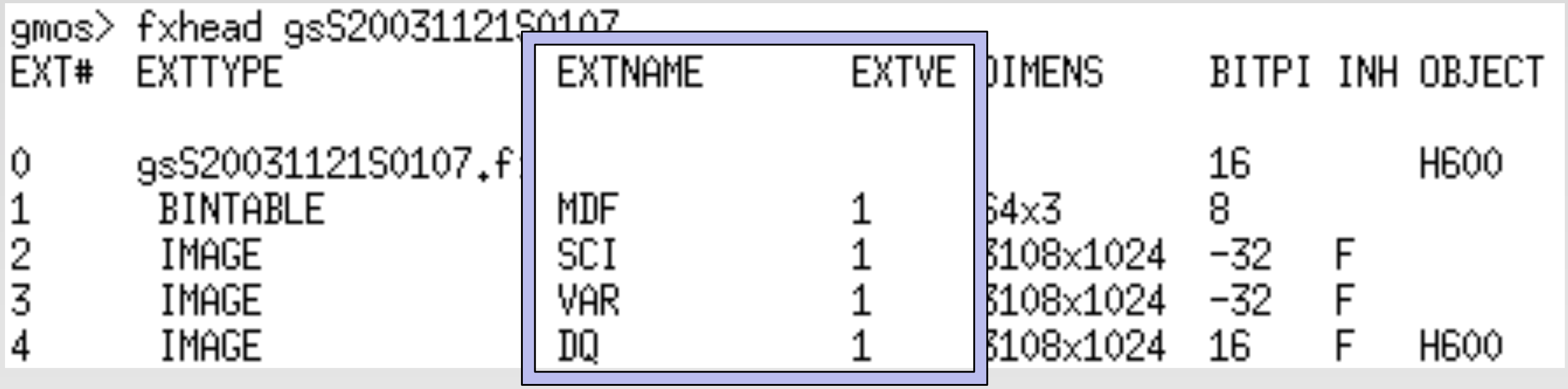


```
gmos> fxhead gS20080220S0078
EXT#  EXTTYPE          EXTNAME        EXTVE  IMENS      BITPI  INH OBJECT

0     gS20080220S0078.fi                                16         ZZ Ori
1      IMAGE           SCI            1      056x512    16     F
2      IMAGE           SCI            2      056x512    16     F
3      IMAGE           SCI            3      056x512    16     F
```

Demo: Gemini Data

➢ For processed GMOS longslit data:



```
gmos> fxhead gsS20031121S0107
EXT#  EXTTYPE            EXTNAME       EXTVE DIMENS     BITPI INH OBJECT

0     gsS20031121S0107.f                                16        H600
1      BINTABLE          MDF           1     64x3       8
2      IMAGE             SCI           1     3108x1024  -32   F
3      IMAGE             VAR           1     3108x1024  -32   F
4      IMAGE             DQ            1     3108x1024  16    F   H600
```

- Specify the extension name and number as the index

```
cl> display gS20110726S0001.fits[SCI,2]
```

- Use `tables.ttools.tread` to read the MDF:

```
cl> tread S20110726S0001.fits[MDF]
```

Demo: Things to know!

- Want to work with text and simple binary files?

```
cl> help system
```

- Want to work with with image files?

```
cl> help images.imutil
```

- The "!" is used as an escape to the host operating system and allows the user to execute a host level command from within IRAF

```
cl> !pwd

cl> !ls
```

GEMINI OBSERVATORY
Exploring the Universe, Sharing its Wonders

Demo: Things to know!

- Image section syntax (x == columns, y == rows)
  - ➢ `[beginning-x:ending-x,beginning-y:ending-y]`
- Not all IRAF tasks allow using image section syntax
- The `imcopy` task does allow it ...
- Copy a section of a FITS file

```
cl> imhead dev$pix
cl> imcopy dev$pix[1:200,1:200] im.fits
cl> display im.fits 1
```

Demo: IRAF Scripting

- Script guide
  - `http://iraf.net/irafdocs/script.pdf`
- Create a script using current IRAF tasks

```
cl> mkscript
```

- Use redirection to execute the script

```
cl> cl < my_task.cl
```

- Or declare a new task (without the $ if task has parameters)

```
cl> task $my_task=/<path>/my_task.cl
```

- Execute the new task

```
cl> my_task
```

GEMINI OBSERVATORY
Exploring the Universe, Sharing its Wonders

Demo: End IRAF session

- Log out of IRAF

```
cl> logout
```