



# I. Configuring and Starting IRAF

[TOC](#) | [previous](#) | [next](#)

---

## Preliminary Notes

As you work with IRAF, you will interact with two command line environments: UNIX and the IRAF CL (CL stands for command line interface). To differentiate between these two command line interfaces, I will prepend a command prompt on the line of any commands I am typing as examples. When working in UNIX, I will use the % as the command line prompt, when working in the IRAF CL, I will use the command prompt `cl>`. Note that the command prompts you see when working on your system may differ. I am assuming for this tutorial, that the reader is already moderately proficient in UNIX.

Also, just like UNIX, IRAF is case sensitive on all filenames and commands. The file `image.fits` is different from `Image.fits` is different from `image.FITS`, so be sure to be careful with upper and lower cases in filenames.

---

## A) Installing IRAF

There are several options for installing IRAF. The original IRAF download page is at [NOAO](#). The install procedure can be a bit complicated if you're not familiar with command line work, fortunately there is another option. The [Scisoft](#) package was created by ESO to provide a standard install for a long list of scientific software (not just IRAF). There are two versions, one for [linux](#) and one for [Mac OS X](#).

There is also a new version of IRAF available (and part of the Scisoft install). IRAF was built by people at NOAO, but another version is being built at STScI called [PyRAF](#). It uses the core IRAF utilities, but makes them available within the Python programming language. PyRAF can be used in two basic ways, in one the IRAF tools are available within the normal Python language. The other is a simulator of the original IRAF command line and works in nearly the same way as IRAF's `cl`.

Because PyRAF combines both the utility of the original IRAF command line and the ability to incorporate IRAF commands in to a modern programming

language, I recommend using PyRAI if available. This tutorial will cover interacting with IRAF through the IRAF command line, but should work almost identically via PyRAI's command line.

---

## B) Using the mkiraf Command

IRAF requires a configuration file (`login.cl`) and a directory to store user parameters (`uparm`). I recommend creating an IRAF directory in your home directory (`~/iraf/`) to store these files. The `mkiraf` command will create the necessary `login.cl` and `uparm` files in the directory in which it is run.

```
% cd ~
% mkdir iraf
% cd iraf
% mkiraf
-- creating a new uparm directory
Terminal types: xgterm,xterm,gterm,vt640,vt100,etc.
Enter terminal type: xgterm
A new LOGIN.CL file has been created in the current directory.
You may wish to review and edit this file to change the defaults.
```

When the `mkiraf` script prompts you for the terminal type, enter `xgterm`. IRAF has several graphical interfaces which you will want to use as you reduce images. These will only run properly if we run IRAF from an `xgterm`. To open an `xgterm` (which behaves in most ways just like a regular `xterm`), use the `xgterm` at the command prompt.

---

## C) Opening an `xgterm`

To open an `xgterm`, type `xgterm` at the UNIX command prompt:

```
% xgterm &
```

You should always run IRAF from within an `xgterm`. Some of IRAF's tasks will use the special properties of an `xgterm` to create interactive graphing windows which you can use to view and manipulate data. Typically you will need only one `xgterm` with IRAF running in it. You can have regular terminals (i.e. an `xterm`) open at the same time, so you can see what files are in the directory you are working in.

---

## D) Editing your `login.cl` File

Before we get started we need to edit the `login.cl` file to reflect a couple of local

preferences for our systems (i.e. cosmos or origins). In UNIX, cd into your `~/iraf/` directory and use your favorite text editor to open up the `login.cl` file. First, find the line that says:

```
#set stdimage      = imt800
```

this is line 27 for me.

This sets the default size of the region displayed in pixels. The pound sign (#) means that this line is commented out and not read by IRAF. Right now, if this command were run (minus the comment sign), the image size would be 800 x 800 pixels. If you are using IRAF on images from a professional observatory, I recommend setting this to 4096, if you're using the camera at SBO, I recommend setting it to 2048. Thus, delete the comment sign and change the 800 to a larger value. For example, in my `login.cl`, this line looks like (note that the # sign has been removed from the beginning of the line):

```
set stdimage      = imt4096
```

The other line in `login.cl` that we need to edit looks like:

```
#set imtype       = "imh"
```

this is line 34 for me.

This sets the default image type. When an image name is expected by a task in IRAF, it will assume that the extension (the letters after the dot in the filename) is of this type unless specified. The `imh` type was developed with IRAF, however most CCD data is now in the FITS (Flexible Image Transport System). Change this line (don't forget to delete the # symbol) to read:

```
set imtype       = "fits"
```

---

## E) Starting IRAF

To start IRAF's command line interface, we use the `cl` command (always remember to start IRAF from an `xgterm`, not a regular `xterm` or terminal). The other thing to remember, is to start IRAF from the directory which contains the `login.cl` script.

```
% cd ~/iraf
% cl
```

```
NOAO PC-IRAF Revision 2.12.1-EXPORT Fri Jul 12 15:54:09 MST 2002
This is the EXPORT version of PC-IRAF V2.12 supporting most PC systems.
```

```
Welcome to IRAF.  To list the available commands, type ? or ??.  To get
detailed information about a command, type `help command'.  To run a
```

command or load a package, type its name. Type `bye' to exit a package, or `logout' to get out of the CL. Type `news' to find out what is new in the version of the system you are using. The following commands or packages are currently defined:

dataio.	language.	noao.	proto.	utilities.
dbms.	lists.	obsolete.	softools.	
images.	mscred.	plot.	system.	

cl>

Now we are interacting with the IRAF command line rather than UNIX. If you forget to start IRAF from the directory which contains the login.cl script, then you will get a message like this:

**Warning: no login.cl found in login directory**

I find it convenient to make a symbolic link to my login.cl file from my home directory, so I can start IRAF from my home directory. To create this link, run the following:

```
% cd ~  
% ln -s ~/iraf/login.cl login.cl
```

Now you can start IRAF from your home directory.

---

[TOC](#) | [previous](#) | [next](#)

---

Copyright © [Josh Walawender](#)