PART 1:

...For All Observers

The chapters in this part of the handbook provide general information about HST data sets and the tools used to work with them. Topics in this part include:

- How to get help and find information.
- How to use IRAF and STSDAS software to work with the datasets.
- What files constitute a dataset.
- How to use the HST Data Archive to find and retrieve files.
- How to understand and use the Observation Log files.

CHAPTER 1: Getting Help and Information

In This Chapter...

In this chapter, we describe the services that STScI provides and how to find and obtain information about the Hubble Space Telescope (HST), its instruments, and observations and their analysis. We also list additional documentation that are available from the Help Desk at STScI.

Additional information about virtually any HST-related topic can be found on the Space Telescope Electronic Information System (STEIS).

Getting Help

User support services are available from either the Space Telescope Science Institute (STScI) or the Space Telescope European Coordinating Facility (ST-ECF). European users should generally contact the ST-ECF staff for help; all other users should contact STScI. If a contact scientist has been assigned to your program, you should address questions to that contact scientist. Otherwise, contact the general Help Desk at STScI; European users can get general help through ST-ECF in Garching.

To contact the Help Desk at STScI:

- **Send e-mail**: `help@stsci.edu`
- **Phone**: 1-410-338-1082

For questions related to the HST Archive:

- **Send e-mail:** `archive@stsci.edu`

To contact the Help Desk at ST-ECF:

- **Send e-mail**: `stdesk@eso.org`

The Help Desk staff at STScI quickly provides answers to any question about HST-related topics. The Help Desk staff has access to all of the experts and resources available at the Institute, and they maintain a database of questions and answers so that frequently asked questions can be immediately answered. The Help Desk staff also provide STScI documentation, in either hardcopy or electronic form, including all user manuals, Instrument Science Reports, Instrument Handbooks, and more).

Questions sent to the Help Desk during normal business hours will normally be answered within that day. Questions received outside business hours will be answered the next morning. Usually, the Help Desk staff will reply with the answer to a question, but occasionally they will need more time to investigate the answer. In these cases, they will reply with an estimated time needed to reply with the full answer.

If you feel that your needs are not being adequately addressed through the Help Desk, please contact the Science Support Division Head, Knox Long, by e-mail at `long@stsci.edu` or by phone (410) 338-4862.

Accessing STEIS

The Space Telescope Electronic Information Service (STEIS) is a collection of electronically available files containing a wide variety of information about HST for professional astronomers and the public at large. Essentially all new documentation created by the Institute is posted to STEIS.

Resources on STEIS includes:

- Information on submitting a proposal for HST observing time.
- Documentation, including instrument handbooks and guides for reducing scientific data.
- News from HST, including recent observations and telescope performance.
- Status of scientific instruments.
- Data reduction software.
- Information about the HST Data Archive.
- The HST observing schedule.
- Calibration information.
- Educational material, including MPEG movies and images of selected HST data (in several popular formats, including TIFF, GIF, and JPEG).[1]

This section describes how to use various access methods to find and retrieve information from STEIS. Most users are now using World Wide Web, which is described in more detail than other methods (such as FTP or Gopher).

World Wide Web

The World Wide Web (WWW) is a hypertext-oriented way of navigating through the Internet and finding information. The WWW project was started by and is maintained by CERN (the European Laboratory for Particle Physics). Hypertext is text that contains links to other documents. Access to WWW is handled by running client software (a *browser*) on your local host. A popular browser is the National Center for Supercomputer Applications (NCSA) Mosaic browser, which will be shown in the examples here. Another popular browser is Netscape.

The browser allows you to read hypertext documents that are made available at individual internet sites (such as STEIS) and to navigate from document to document by fetching the selected hypertext documents. The location of each document is described by an address called (in WWW jargon) the *uniform resource locator* (URL). The home page for STEIS (Figure 1.1 shows the STEIS *home page*), is fetched by the following URL:

```
http://www.stsci.edu/top.html
```

You could also give a URL when you first start Mosaic, for example:

```
Mosaic http://www.stsci.edu/top.html
```

After starting Mosaic, you can specify a URL by clicking on the "Open" button and typing the URL in the pop-up window. The highlighted text is connected or *linked* to other documents, images, or files. To navigate in hyperspace just click on the highlighted text with the left mouse button and you will be connected to that document. You can save a document to disk by clicking the "Save As" button and a pop-up window will appear. You can specify a directory and file name for the file and choose a format, i.e., ASCII, PostScript, or Hypertext Markup Language (HTML). (In Netscape, you can do this by holding down the right mouse button over a link.) Similar to Gopher's bookmarks, it is possible to save in a *hotlist* the locations of documents that you may wish to revisit later. Click on the "Navigate" button at the top of the screen and in the pull-down menu click on "Add current to hotlist". Clicking on "hotlist" in the same pull down window allows you to select and fetch the documents. In Netscape, the equivalent feature is called "bookmarks" and is available from the menu bar.

---

1. MPEG is the Motion Picture Encoding Group format. TIFF is Tagged Image File Format. GIF is Graphic Interchange Format. JPEG is Joint Picture Encoding Group format. Each of these are widely-understood graphic file formats.

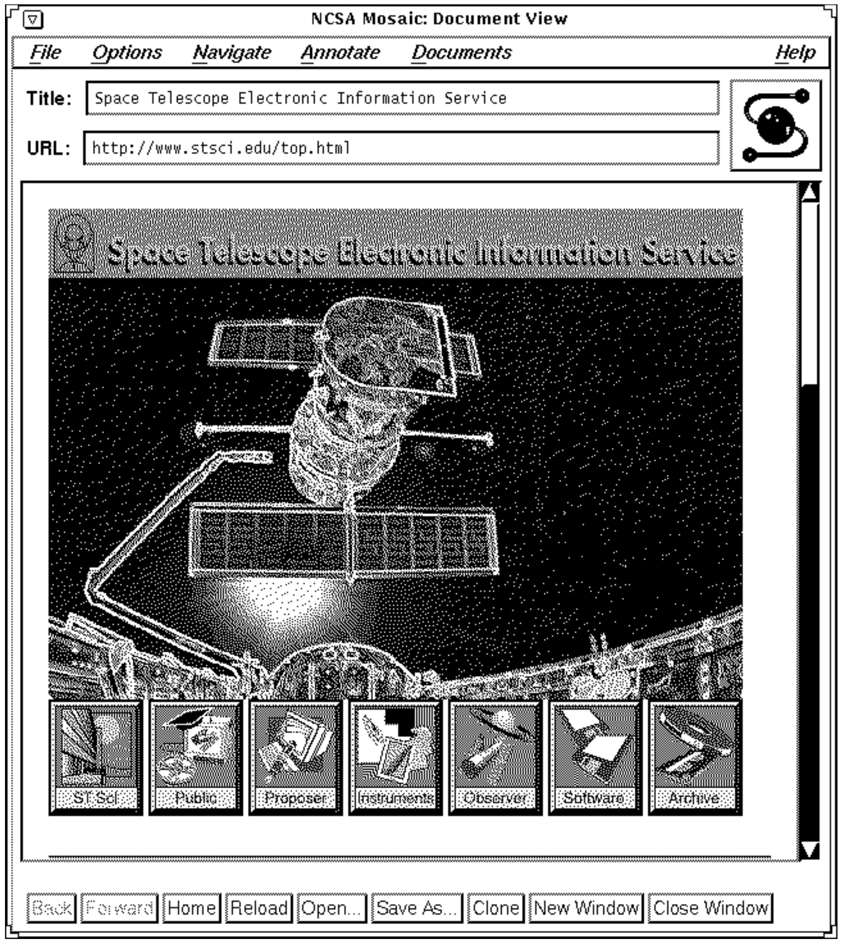


Figure 1.1: STEIS Home Page

World Wide Web is the recommended method for obtaining electronic information from STEIS. As you can see in Figure 1.1, the home page provides links to seven major areas:

- ***STScI***: Information about the Space Telescope Science Institute, its staff, library, and meetings, and about the Publications of the Astronomical Society of the Pacific (PASP).
- ***Public***: Contains links to information of general interest to educators, students, the media, and the general public. These include pictures (in GIF, TIFF, and JPEG formats) obtains from HST observations, movies and animation, and the text from press releases. It also includes HyperCard books for the Macintosh.
- ***Proposer***: Information and tools for astronomers proposing HST observations. Resources include the call for proposals, templates and instructions, and documentation.
- ***Instruments***: Information on the scientific instruments aboard HST. Includes links to handbooks, instrument status, calibration information, and Instrument Science Reports.

- ***Observer***: Information and resources for HST observers, including Phase II proposal and budget templates and software, as well as the weekly HST observation schedule.
- ***Software***: Contains links to the STSDAS software, Digitized Sky Survey CD-ROM information, **hcompress** image compression software, and Telescope Image Modeling software.
- ***Archive***: Information and tools for searching and using the Archive of past HST observations.

Structure

Table 1.1 shows the major directories on STEIS[2] and describes their contents. Those directories in the white bars contain files and subdirectories mentioned in this document. The shaded rows contain information that is unlikely to be needed to reduce or analyze HST datasets. This structure can be expected to change over time.

Some of the resources available on STEIS that will be of interest to observers analyzing HST data include:

- Calibration reference files in the `cdbs/` directory.
- *Instrument Handbooks* in the `documents/` directory.
- The *HST Archive Manual* in the `documents/` directory.
- Gopher access (via telnet) to the HST Archive host computers through the `hst-archive/` directory.
- STEIS memos mentioned in Book II and other instrument information are in the `instrument_news/` directory.
- STSDAS software and documentation can be retrieved from the `software/` directory.
- TIM and TinyTIM software are in the `software/` directory.

2. Current as of December 1995.

Table 1.1: STEIS Directory Structure (Shaded blocks are not critical))

Directory	Description	Major Files and Subdirectories
cdbs	Files used in recalibrating HST data.	calobs/ calspec/ comp/ crwave/ fields/ grid/ uref/ utab/ vtab/ wref/ wtab/ yref/ ytab/ zref/ ztab/
documents	ASCII and PostScript versions of documents.	archive-manual/ baltimore-charter/ calibration/ fgs-handbook/ foc-handbook/ fos-handbook/ gasp-cookbook/ ghrs-handbook/ hsp-handbook/ image-restoration/ phase2-instructions/ wfpc2-handbook/ network-resources/ stsdas-docs/
ExInEx/	HyperCard images from the Special Studies Office.	
hst-archive/	Files and tools of use to archival researchers. If using WWW or Gopher, it is possible to access the HST Data Archive from within this directory.	
instrument-news/	Abstracts of most recent instrument reports written by instrument scientists, describing reduction techniques, etc.	fgs/ foc/ ghrs/ hsp/ observatory/ ota/ scs/ smo/ wfpc/ wfpc2/
meetings/	Information about upcoming meetings at STScI.	
observer/	Information of intereste to observers, including catalogs and schedules of upcoming observations.	/catalogs/ weekly_timelines/
pasp/	Information from Publications of the Astronomical Society of the Pacific.	
policy/	Copies of letters to PIs, membership on STScI committees, STScI long-range plans.	
proposer/	Information and resources to propose HST observations.	cycle6/ software/
software/	Software for use with HST data, including STSDAS, TIM, and image compression software.	cdrom/ casb/ hcompress/ rps2/ stsdas/ tim/
stsci/	Information for and about STScI, includes images in epa directory, information about grants, and other Institute news.	epa/ grants_office/ hst_news/ human_resources/ library/ newsletters/ steis/

Each instrument team maintains informational listings about currently available reference files on STEIS. These listings often contain information about the nature and quality of the reference files. You may wish to consult these listings before you decide to recalibrate. Each of the instrument chapters will describe necessary reference files and provide information about locating and retrieving the files.

Gopher

You can also access STEIS through Gopher if the software is installed at your home site. Gopher is a user-friendly interface developed at the University of Minnesota; it allows you to browse through files before retrieving them and allows you to search WAIS indexes. If you have Gopher installed, you can access

STEIS by typing one of the following commands. Figure 1.1 shows how STEIS looks from within Gopher.

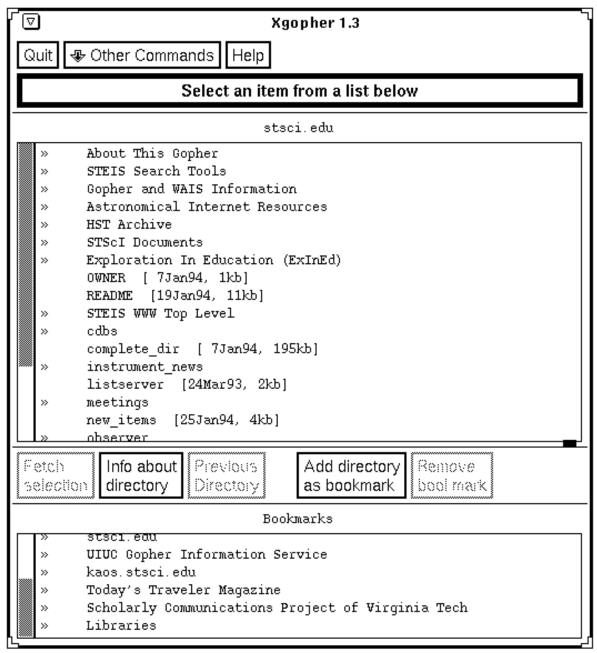
Terminal-Based Gopher:

```
gopher www.stsci.edu
```

X-Windows Gopher:

```
xgopher www.stsci.edu
```

Figure 1.1: Xgopher



Anonymous FTP

You can log in to STEIS by using FTP to open a connection to `ftp.stsci.edu` and then log in with username "anonymous" and your e-mail address as a password. Use the `cd` command to change directory (e.g., `cd /instrument_news`), `ls` to get a listing of the contents of a directory, and `get filename` to transfer a single file to your home computer via FTP. Use `mget` (e.g., `mget template*`) to get a series of files. Images and tables should be transferred in binary mode. Type `binary` at the ftp prompt before getting binary files. If you find files with a `.Z` extension, the file is compressed. If you leave off the `.Z` extension when typing your `get` command, the server will automatically uncompress the file for you.

Most images on STEIS are Unix format GEIS files and most tables are Unix format STSDAS tables. If you wish to use these files on a VMS computer you will need to convert the images and tables to VMS format using **sun2vax** and **tconvert** (both tools within STSDAS), respectively.

Listserver

Users can subscribe to certain popular files, which will be e-mailed automatically whenever they're updated. It is also possible to request items from the archive of previous files. The list of available files will expand with time to include other frequently accessed ASCII files. Please send any requests for additional files you would like to see made available through listserver to `hst_query@stsci.edu`. Files currently available for subscription are listed in Table 1.1.

Table 1.1: Files Available Through Listserver

File Name	Description
`timeline`	Detailed weekly schedule of observations
`hst-status`	Daily activity and instrument status reports
`wfpc2_cal`	Calibration information for WFPC2
`foc_news`	Information about the Faint Object Camera
`fos_news`	Information about the Faint Object Spectrograph
`ghrs_news`	Information about the Goddard High Resolution Spectrograph
`wfpc_news`	Information about both the WF/PC-1 and the WFPC2

To subscribe to a listserver file, send a message with a blank subject line to `listserv@stsci.edu` and the following text:

`subscribe` *file name* *Your Name*

where "file name" is one of the above file names and "Your Name" represents the user's full name. Your return e-mail address is copied from this message and added to the list of subscribers. To unsubscribe from a list, just replace the word "subscribe" with "unsubscribe" in the above message.

For a quick overview of commands, send a message with a blank subject line to the same e-mail address but with the word "HELP" as the message.

Additional Documentation

STScI maintains a complete list of available documentation. Copies of this list can be obtained by sending an e-mail request to `help@stsci.edu`. Some of the documentation that expands on information provided in the *HST Data Handbook* include:

- *STSDAS Users Guide,* version 1.3, February 1994.
- *HST Archive Primer*, version 5.1, May 1995.
- *HST Archive Users Manual*, version 5.1, February 1995.
- *FOC Instrument Handbook*, version 6.0, June 1995.
- *FOS Instrument Handbook*, version 6.0, June 1995.
- *FGS Instrument Handbook*, version 5.0, June 1995.
- *GHRS Instrument Handbook*, version 6.0, June 1995.
- *WF/PC-1 Instrument Handbook*, version 5.0, April 1994.
- *WFPC2 Instrument Handbook*, version 6.0, June 1995.
- *Synphot User's Guide,* version 1.3.3, March 1995.
- *Calibration of the Hubble Space Telescope: Proceedings from a Workshop Held at STScI November 1993*, J. Chris Blades and Samantha Osmer (eds.).
- Various Instrument Science Reports describing technical and calibration issues.

CHAPTER 2:
Data Analysis with IRAF and STSDAS

In This Chapter...

The Space Telescope Science Data Analysis System (STSDAS) is the software system for calibrating and analyzing data from the Hubble Space Telescope. The package is large and powerful, containing programs—called *tasks*—to perform a wide range of functions supporting the entire data analysis process, from reading your tapes, through recalibration and analysis, to producing your final plots and images.

STSDAS is built on top of the Image Reduction and Analysis Facility (IRAF) software developed at the National Optical Astronomy Observatory (NOAO). This means that any task in IRAF can be used in STSDAS and that the software will be portable across a number of platforms and operating systems. To effectively exploit the power of STSDAS, you will need to learn the basics of IRAF.

IRAF Primer

This section provides just enough information to help you start effectively using IRAF and STSDAS. These concepts are general, and apply to using tasks in IRAF, STSDAS, TABLES, or any IRAF layered package. As your experience with the software increases, you will need to reference more complete information, which can be found in the *STSDAS User's Guide* and in the documentation available from NOAO, especially *A Beginner's Guide to Using IRAF*. Some of the concepts described in this chapter include:

- How to set up IRAF the first time you use the software.
- How to start and stop an IRAF session.
- Basic concepts, such as loading packages, setting parameters, etc.
- How to use the on-line help facility.
- File formats used in IRAF and STSDAS, such as Generic Edited Information Set (GEIS) image format and STSDAS table format.

First Time...

This section is for new IRAF users. It explains:

- How to set up your working IRAF environment (below).
- How to start the IRAF program and how to quit when you're done (page 16).

We assume that your site has IRAF and STSDAS installed. If this is not the case, you must obtain and install the software. See "Getting IRAF and STSDAS" on page 70 for details.

Setting Up IRAF

Before running IRAF for the *first time*, you must:

1. Define your root IRAF directory.
2. Define environment variables or system logicals and symbols.
3. Use **mkiraf.**

First, you must decide which directory you will use as your *root* IRAF directory (also referred to as your IRAF *home* directory). Users generally name their IRAF home directory `iraf` and set it up in their account's root directory (i.e., the default directory that you are in when you log in to the system). The IRAF home directory doesn't need to be in your account's root directory, nor does it need to be called `iraf`; it can be anywhere and be named anything, but you should *not* put it on a scratch disk that is periodically erased.

After you define your home directory, switch to that directory so that you can define the needed environment variables or system logicals and symbols. Assuming that you call your root IRAF directory "IRAF", this is done as follows:

For VMS:

Can be placed in LOGIN.COM file

```
$ CREATE/DIR [.IRAF]
$ SET DEFAULT [.IRAF]
$ IRAF
$ MKIRAF
```

For Unix:

Can be placed in .login file

The directory name is site-dependent—check with your system staff

```
% mkdir iraf
% cd iraf
% setenv iraf /usr/stsci/iraf
% source $iraf/unix/hlib/irafuser.csh
% mkiraf
```

The **mkiraf** command initializes IRAF. The command works the same under either VMS or Unix. The **mkiraf** command does two things for you:

- Creates and initializes your `login.cl` file.
- Creates a subdirectory called `uparm` .

After typing the **mkiraf** command, you will see the following:

```
% mkiraf
-- creating a new uparm directory
Terminal types: gterm=ttysw+graphics,vt640...
Enter terminal type:
```

Enter the type of terminal or workstation you will most often use with IRAF. Terminal types and workstation emulators will vary from site to site and you should look at the `dev$termcap` file for a complete list of terminal types.[1] Generic terminal types that will work for most users are:

- `vt100` for most terminals.
- `xtermjhs` for most workstations running under X-Windows.
- `xgterm` for sites that have installed X11 IRAF and IRAF v2.10.3 BETA or later.

You can change your terminal type at any time by typing `set term=`*new_type* during an IRAF session. You can also change your default type by editing the appropriate line in your `login.cl` file.

After you enter your terminal type, you will see the following output before getting your regular prompt:

1. Users at STScI should consult the *STScI Site Guide for IRAF and STSDAS*.

```
A new LOGIN.CL file has been created in the current ...
You may wish to review and edit this file to change ...
```

The `login.cl` file is the *startup file* used by the IRAF command language (CL). It is similar to the `LOGIN.COM` file used by VMS or the `.login` file used by Unix. Whenever IRAF starts, it looks at the `login.cl` file. You can edit this file to customize your IRAF environment. In fact, you should look at it to make sure that everything in it is correct. In particular, there is a line starting with `set home =` that tells IRAF where to find your IRAF home directory. You should verify that this statement does, in fact, point to your IRAF directory. If you will be working with IRAF format images (OIF) you should also insert a line saying `set imdir = "HDR$"`. The `imdir` setting is ignored when working with GEIS format images.

The `uparm` directory will contain your own copies of IRAF task parameters. This allows you to further customize your IRAF environment by setting certain parameter values as defaults. Once you set up IRAF, you should rarely need to do it again. When you want to use IRAF, simply move to your IRAF directory and type `cl`.

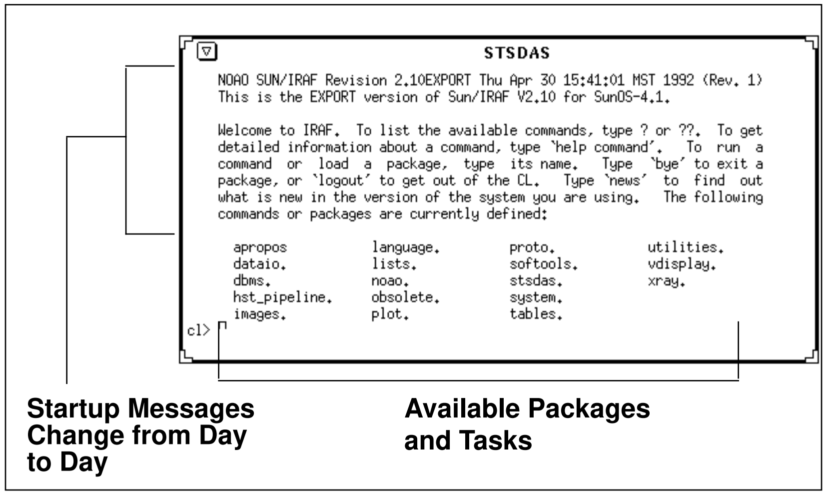
Starting and Stopping an IRAF Session

To start an IRAF session:

1. Move to your IRAF home directory.
2. Type `cl`.

   IRAF starts by displaying several lines of introductory text and then puts a prompt at the bottom of the screen. Figure 2.1 is a sample IRAF startup screen.

Figure 2.1: IRAF Startup Screen



To quit an IRAF session:

1. Type `logout`.

IRAF Concepts

This section describes basic techniques such as:

- Loading packages (below).
- Running tasks and commands (page 19).
- Viewing and setting parameters (page 19).
- Setting and using environment variables (page 22).

The typical sequence of doing things in IRAF is:

1. Define any environment variables that you may want to use during your session, such as the location of reference files and tables.

2. Load the packages you want to use.

3. Edit parameter sets for the tasks you want to use.

4. Run the tasks.

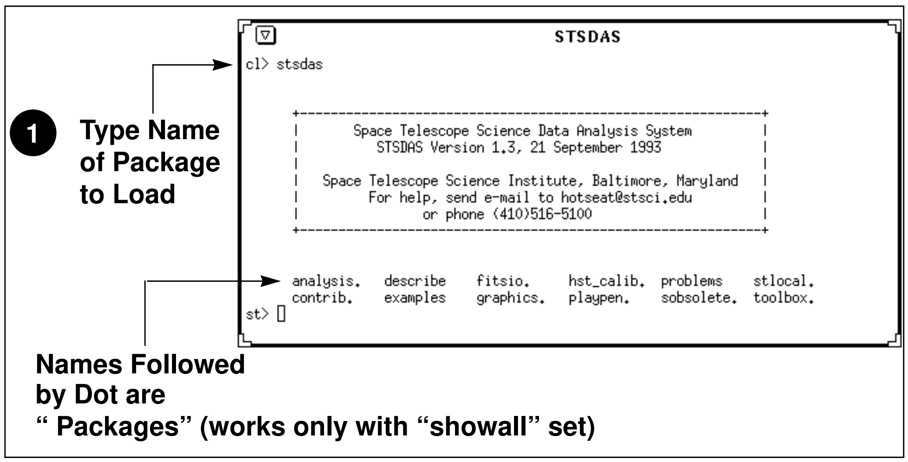
Each of these steps are described in the following sections, although not in this order because the most fundamental and commonly-used concepts are described first. For example, not every user will need to redefine environment variables, especially during a first session.

Loading Packages

In IRAF jargon, an application is called a *task* and logically related tasks are grouped together in a *package*. Before you can use a task, its package must be loaded.

- Load a package by typing its name.
- When you load a package, the prompt changes to the first two letters of the package name.
- When you load a package, the names of all the newly-available tasks and subpackages are displayed.

Figure 2.2: Loading Packages



Some commands that will help you manage packages are:

- `?` - List tasks in the most recently-loaded package.
- `??` - List all tasks loaded, regardless of package.
- `package` - List names of all loaded packages.
- `bye` - Exit the current package.

Tasks and Commands

This section explains how to run tasks, run background tasks, run system-level commands, and use piping and redirection.

Running a Task

- Run a task by typing its name, or any unambiguous abbreviation of it.

Normally, you would also specify values for any required *parameters* on the command line when you run the task, however, you don't need to. If you start a task by simply typing its name, the task will prompt you for any required information, such as the names of input files, starting values, etc.

The use of parameters is an important concept and will be discussed in the next section. An example of running a task with a single parameter is:

```
st> gstatistics dev$pix
```

IRAF does not require you to type the complete command name—only enough of it to make it unique. For example, **dir** is sufficient for **directory**.

Background Tasks

To run a task as a background job, freeing your workstation window for other work, add an ampersand (&) to the end of the command line. For example:

`st>` *taskname* `&`

Escaping System-Level Commands

To run an operating system-level command (i.e., Unix or VMS commands) from within the IRAF CL, precede the command with an exclamation point (!). This is called *escaping* the command. For example:

`st> !`*system_command*

Piping and Redirection

You can run the tasks in sequence, with the output of one task being used as the input for another. This is called *piping*, and is done by separating commands with a vertical bar (|). For example:

`st>` *command filename* `|` *task2*

You can also redirect output from any task or command to a file by using the greater-than symbol (>). For example:

`st>` *command* `>` *outputfile*

Setting Parameters

Information is provided to tasks through *parameters*. Parameters are used to specify names of input or output files, starting or ending values, option settings, and many other types of information that controls the behavior of the task.

The two most useful commands for handling parameters are:

- **lparam** to display the current parameter settings.
- **eparam** to edit parameters (page 20).

Parameters can be set in three ways:

- From the command line. For example, you could type:

`st>` *task.parameter="value"*

- When running the task. For example:

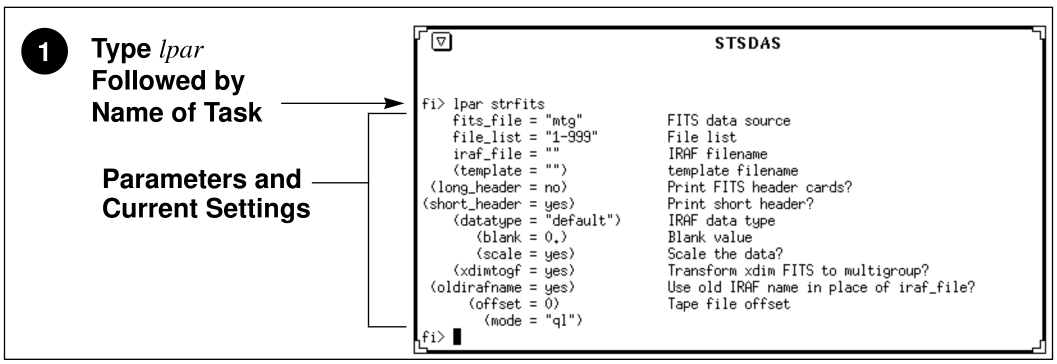
`st>` *task parameter_value*

- Using **epar**.

Viewing Parameters with lparam

The **lpar** command lists the current parameter settings for a given task (Figure 2.3).

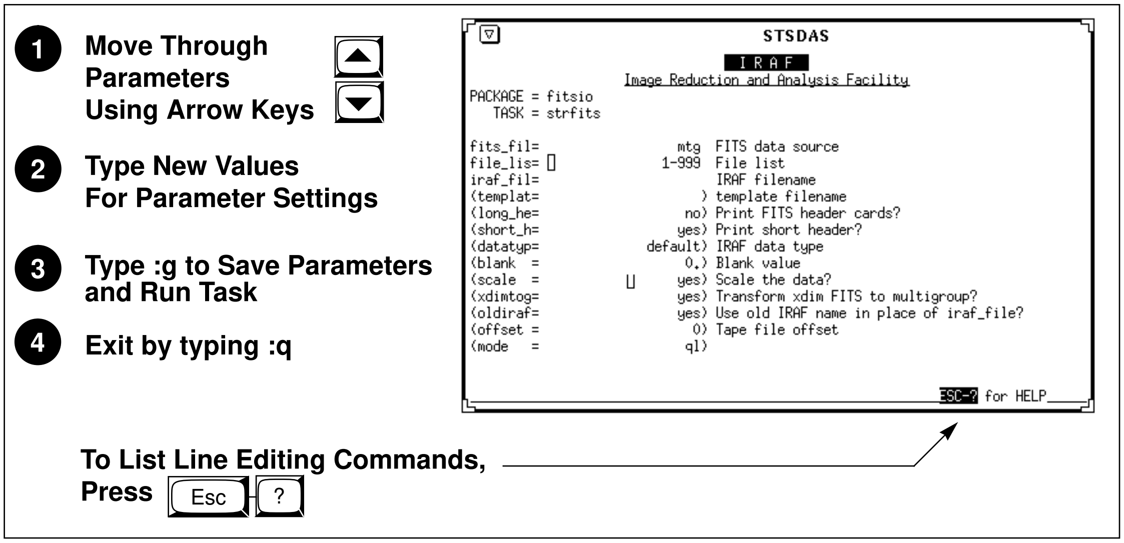
Figure 2.3: Displaying Parameter Settings with lpar



Setting Parameters with eparam

The **epar** command is an interactive parameter set editor. All of the parameters and their current settings are displayed on the screen and you can move around the screen using the arrow keys (also called *cursor* keys) and type new settings for any parameters you wish to change. Figure 2.4 shows a sample of the **epar** editor at work (invoked by typing `epar strfits`).

Figure 2.4: Editing Parameters with epar



Parameter Data Types—What to Specify

Parameters are either *required* or *hidden*, and each parameter expects information of a certain *type*. Usually, the first parameter is required, and very often it expects a file name. Parameters are described in the online help for each task. When a parameter is shown in parentheses (either in the online help or in the **lpar** listing), the parameter is optional.

If you type the wrong type information for a parameter, **epar** has some error checking capability and will usually display an error message saying “Parameter Value is Out of Range.” The message is displayed when you move to another parameter or if you press Return. Parameter types are listed in Table 2.1.

Table 2.1: Parameter Data Types

Type	Description
`File Name`	Full name of the file. Wild card characters (* and ?) are often allowed. Some tasks will allow you to use special features when specifying file names, including “@” lists, IRAF networking syntax, and image section or group syntax. (See “File Name Syntax” below).
`Integer`	Whole number. Often the task will specify minimum or maximum values (see the help pages).
`Real`	Floating point numbers, can be expressed in exponential notation. Often will have minimum and maximum values.
`Boolean`	Logical “yes” or “no” values.
`String`	Any characters. Sometimes file names are specified as string.
`Pset`	Parameter set.

Restoring Parameter Default Values

Occasionally, IRAF will get confused by your parameter values. If this happens, you can restore the default parameters with the **unlearn** command. You can use **unlearn** on either a task or on an entire package.

The **unlearn** command can be used if you have problem with IRAF or STSDAS parameter conflicts.

Setting Environment Variables

IRAF uses *environment variables* to define which devices are used for certain operations. For example, your terminal type, default printer, and the disk and directory used for storing images are all defined through environment variables. Environment variables are set using the **set** command and are displayed using the **show** command. Table 2.2 lists some of the environment variables that you might want to change.

Table 2.2: Environment Variables

Variable	Description	Example of Setting
`printer`	Default printer for text	`set printer = lp2`
`terminal`	Terminal type	`set term = xterm`
`stdplot`	Default printer for all graphics output, such as snap	`set stdplot = ps2`
`stdimage`	Default terminal display setting for image output (most users will want this set to either imt512 or imt800)	`set stdimage = imt800`
`clobber`	Allow or prevent overwriting of files	`set clobber = yes`
`imtype`	Default image type for output images. “imh” is old IRAF format, “hhh” is STSDAS GEIS format.	`set imtype = “hhh”`

You can permanently set your environment variables by editing your `login.cl` file so that they will automatically be set each time you use IRAF. To do this, you would use vi, Emacs, or whatever your favorite text editor happens to be, to specify each variable on its own line. You can see the name and current value of all environment variables for your session by using the **show** command with no arguments.

Working with Files

This section describes:

- File structures commonly used in STSDAS and IRAF.
- How to use directory specifications.
- How to specify file names.
- How to write files to tape using **stwfits**.

File Structures

IRAF recognizes a number of different file structures. HST data are stored in a specific file format known as GEIS, which is different from the IRAF image format. Both IRAF and STSDAS images consist of two files. The two files are *always* used together as a pair:

- A *header file*, which consists of descriptive information. IRAF header files are identified by an extension of `.imh`. STSDAS header files are in ASCII text format and are identified by an extension of `.hhh` or another extension ending in “h”, such as `.c0h` or `.q1h`.
- A *binary data file*,[2] consisting of pixel information. IRAF data file names end with a `.pix` extension. STSDAS data files end with an extension of `.hhd` or another extension that ends with “d”, such as `.c0d` or `.q0d`.

STSDAS always expects both component files of a GEIS image to be kept together in the same directory.

When working with IRAF or STSDAS images, you need only specify the header file name—the tasks will automatically use the binary data file when necessary.

IRAF and STSDAS image formats are not similar, other than that they each have two component files. The IRAF header is part binary and part ASCII, and has no fixed format, whereas the STSDAS header is always 80 bytes long and consists entirely of ASCII text. The IRAF binary file consists of a grid of binary data preceded by a 512-byte binary description. The STSDAS binary file consists of a series of groups of binary data, each representing an entire image, and blocks of group description information called the *Group Parameter Block*.

Directory Specification

As we mentioned above, the two components of an image are expected to be in the same directory, and will look for these files in whatever directory you are currently located.[3]

To navigate through directories, you should use the following commands:

2. The binary data file format is host-dependent and may require translation before it can be moved to a computer using a different architecture.

3. IRAF format images can use separate directories for the header and pixel files.

- **path** - Lists the current working directory.
- **cd** *directory* - Move to the named directory.

Specifying Files

Most tasks in IRAF and STSDAS operate on files and expect you to specify a file name for one or more parameters. There are several types of special syntax that can be used with certain tasks when specifying file names. These syntax features include:

- **Wild card characters**, often called *templates*: These are used to specify multiple files using pattern matching techniques. The wild cards are:
  - * Matches any number of characters, e.g.: `z*.c0h`
  - ? Matches any single character, e.g.: `z01x23x.c?h`

> When using wildcards with image-processing tasks, be sure to exclude the binary pixel files by ending your file name specification with an "h", for example: `y*.??h`

- **List files**, often called *@-files*: A list file is an ASCII file that contains a list of file names, one per line. If the task supports the list file feature, you would type the name of your list file, preceded by the "@" character. For example: `@files.txt`
- **Image section** specification. Tasks that work with image data will often let you specify that you want to work on only a small area of the image rather than the entire image. See "Image Display with the IRAF display Task" on page 29. To specify an image section, specify each axis range in square brackets, for example: `image.hhh[10:200,20:200]`
- **Group** specification. HST images are stored in a format known as *GEIS format*. This lets one file contain one or more sets of related data, such as images taken from different detectors or data taken at different time steps. Many STSDAS tasks will let you choose which of the groups in a multi-group image will be used. To specify a group, enclose its number in brackets, for example: `image.hhh[7]`. A description of GEIS format is available on-line within STSDAS by typing `help geis`. Note that if you use both a group and an image section specification, the group *must be specified first*, for example, `image.hhh[2][10:200]`.
- **IRAF networking** specification. IRAF is capable of reading and writing files to and from remote systems on a network. This is often used with tasks in the **fitsio** and **convfile** packages, or with image display tasks. You must do some work to enable this feature, however, and this is described in the *STSDAS Users Guide* and in the online help, which can be read by typing `help networking`. To specify that you want to use the IRAF networking feature, you will specify the remote host name followed by an exclamation point (!), followed by the file or device name. For example: `ra!mta`.

Writing Files to Tape

The **stwfits** task (in the STSDAS **fitsio** package) is used to create FITS tape (or disk) files from multigroup HST images. The task offers several options for different ways of writing the files, but we will focus here on using the task to write files using the standard IEEE FITS format.[4]

There are two ways to write HST images to FITS files using the **stwfits** task:

- With the groups stacked in a single FITS file and group parameters stored in a FITS TABLE extension.
- With each group written to a separate FITS file.

The first method has the advantage of being somewhat more efficient, but is a relatively recent method of writing data, so older versions of some software may not recognize the TABLE extension. Writing each group of a GEIS file out to a separate FITS files may, therefore, give you better portability to other systems and software.

To write a series of GEIS images corresponding to an HST dataset out to FITS format using the stacking method with a FITS TABLE extension, you could use a command like the following:

```
fi> stwfits z03*.??h mta gftoxdim+
```

To write the same set of files to a FITS tape with each group in a separate image, you could use:

```
fi> stwfits z03*.??h mta gftoxdim-
```

VAX users should be aware that you may lose one digit of precision when using IEEE format because the VAX real*8 format uses 3 bits more than IEEE real*8. Under DEC OpenVMS with the new AXP architecture the bit counts are the same and there is no loss of precision.

On-Line Help

This section describes:

- How to use IRAF's on-line help facility.
- How to find a task that does what you want (page 26).

If you need additional help, a table of contact information is provided in the preface to this manual.

Getting Help

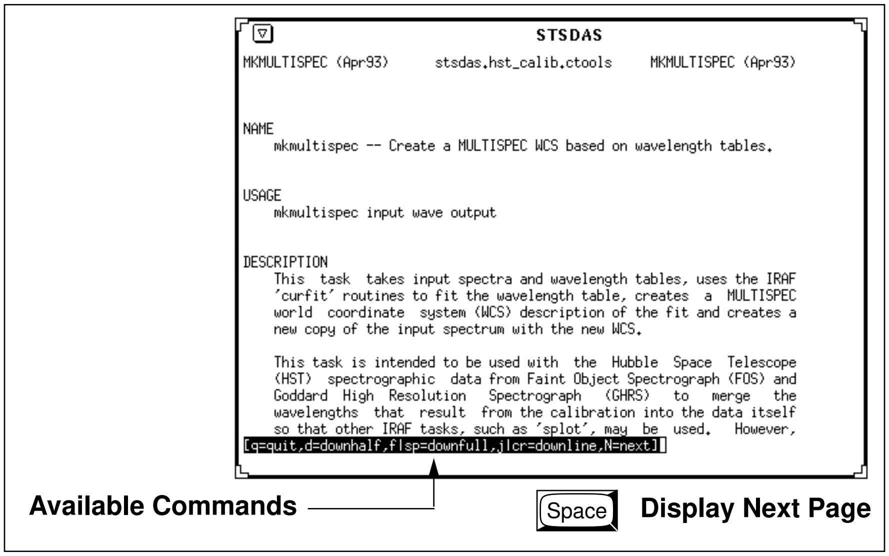
On-line help is displayed by using the **help** command,[5] which takes as an argument the task or package name about which you want help. Wildcards are

4. If you want to know how to use other formats, type `help fits_exampl` from within IRAF.

supported. For example, to display the on-line help for the **mkmultispec** task, you would type:

```
fi> help mkmultispec
```

Figure 2.5: Displaying On-line Help



Two STSDAS tasks are available that will display only certain sections of the help file:

- **examples** - Displays only the examples for a task.
- **describe** - Displays only the description.

Finding Tasks

There are three ways to find a task that does what you need:

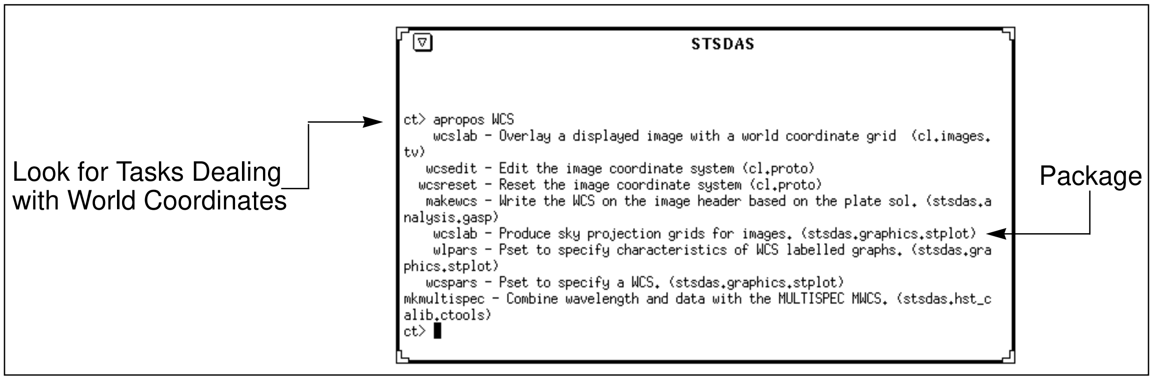
- Use the **apropos** or **references** task to search the online help database.
- Look at the package structure to find the right package.
- Typing `help` *package* will list one-line descriptions of each task in the package.
- Ask. A more experienced user can usually point you in the right direction. You can also request help from the Help Desk staff at STScI by sending e-mail to: `help@stsci.edu`

Using apropos

The **apropos** task looks through a list of IRAF and STSDAS package menus to find tasks that match a specified keyword. Figure 2.6 shows how to use **apropos**. Note that the name of the package in which the task is found is shown in parentheses.

5. There is an optional *paging* front-end for help called **phelp**. For more information, type `help phelp` from within IRAF.

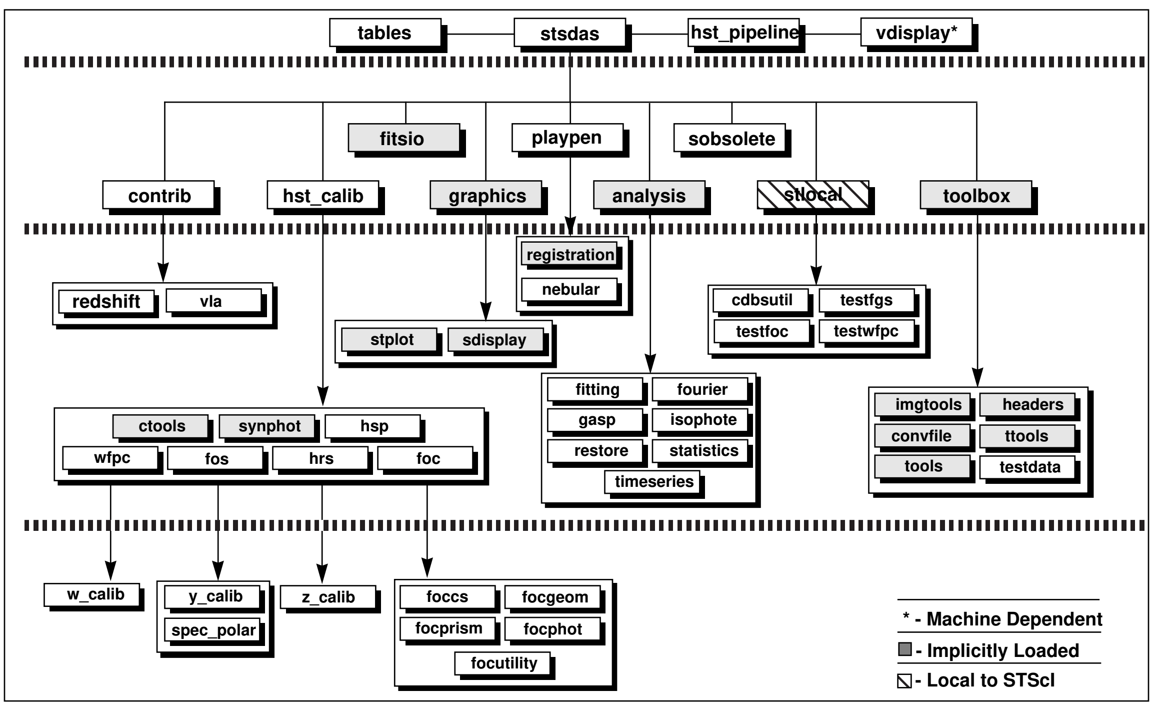
Figure 2.6: Using apropos



STSDAS Structure

STSDAS version 1.3 is structured so that tasks are logically grouped. In many cases, you can easily find a task that does whatever function you need simply by looking in the appropriate package. For example, all of the tasks that are used in the calibration process can be found in the **hst_calib** package and all tasks used for image display and plotting can be found in the **graphics** package. Figure 2.7 shows the STSDAS package structure.

Figure 2.7: STSDAS Version 1.3 Package Structure



Troubleshooting

There are a couple of easy things you can do to make sure that you don't have a simple memory or parameter conflict—common causes of problems.

- Look at the parameter settings and make sure that you have specified reasonable values for every parameter.
- Use the **flprcache** command to clear the process cache. To do this type: `flpr` You may need to do this twice in succession.
- Occasionally, you may need to logout of the CL, restart IRAF, and try your command again.

If you still have a problem, contact the Help Desk. E-mail addresses and phone numbers are provided in the Preface of this manual.

STSDAS Tables

Several of the analysis packages in STSDAS create output files in STSDAS table format, which is a binary row-column format (ASCII-format tables are also supported). You may also find STSDAS tables on your HST data tape since all instruments other than the WF/PC-1 use STSDAS-format tables in the calibration process.

Tables can be viewed, created, and manipulated using the tasks in the **ttools** package (or in the external TABLES package). You can read a description of the table data format in the *STSDAS User's Guide*.

To simply read the contents of a table, you can:

- Use **tread** to display the file and move through it using the arrow keys.
- Use **tprint** to display the file in a formatted manner.

Other tasks are available in **ttools** that will let you edit tables or create new ones, or that will let you manipulate the data, convert formats, or perform various database functions with the tables. See the online help for details.

Other Topics

Other Image Tools...

Both IRAF and STSDAS contain a huge number of tasks that work with images. Some of the packages that you should investigate are:

- **images**: This package includes general tasks for copying (**imcopy**), moving (**imrename**), and deleting (**imdelete**) image files. These tasks operate on both the header and data portions of the image. The package also contains a number of general purpose tasks for operations such as rotating and magnifying images.
- **stsdas.toolbox.imgtools**: This package includes general tools for working with multigroup GEIS images. This package includes tasks for working with masks, and general purpose tasks for working with the pixel data, such as an interactive pixel editor (**pixedit**).
- **stsdas.analysis**: General tasks for image analysis.

Displaying HST Images

This section will primarily be of interest to WF/PC-1, WFPC2, and FOC observers, it explains:

- How to display images in IRAF.
- Some tips for working with HST multigroup format images (page 31).
- How to combine the four groups of a WF/PC-1 or WFPC2 image into a single mosaic image using either the STSDAS **wmosaic** or **qwmosaic** tasks (page 32).

Image Display with the IRAF display Task

The most general IRAF task for displaying image data is the **display** task. This is the best choice for a first look at FOC, WF/PC-1, or WFPC2 data, but it can also be used to display HSP area mode images. To display an image, you need to:

1. Start an image display server (we use SAOimage) in a separate window from your IRAF session (i.e., either from a different xterm window, or as a background job before starting IRAF).[6] To start SAOimage, type the following in any xterm or other system window:

```
saoimage &
```

2. Load the **images** and **tv** packages (from the window where you're running IRAF):

```
cl> images
im> tv
```

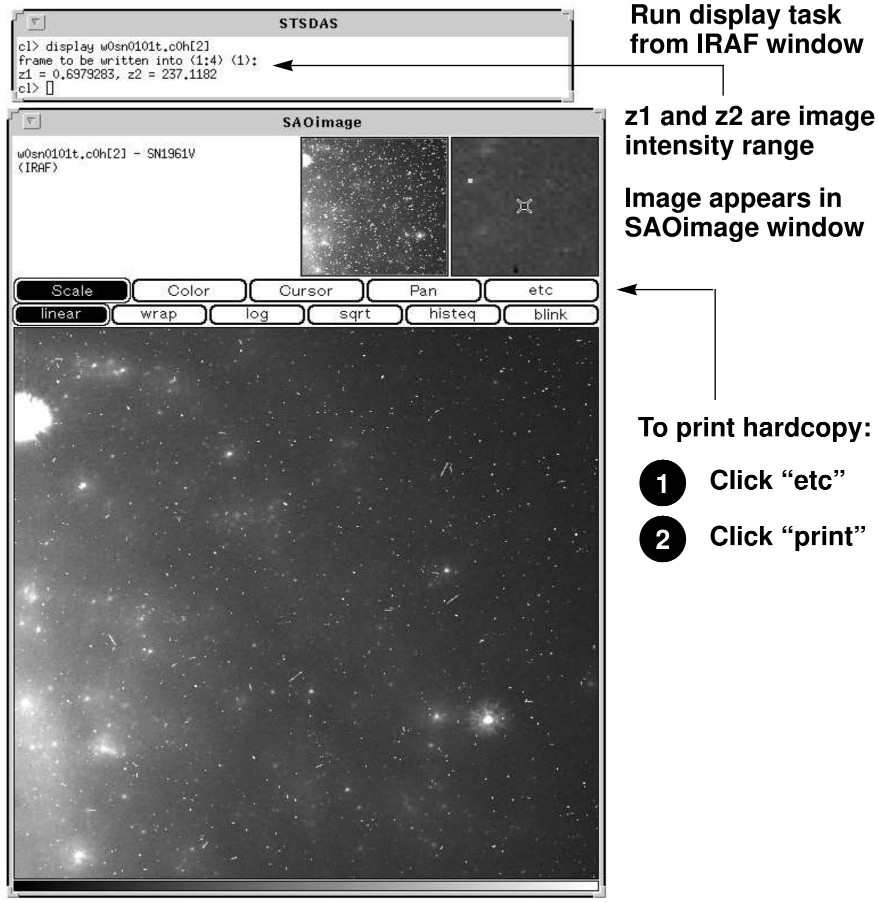
3. Display the image using the IRAF **display** task:

tv> display *rootname.c1h*

Figure 2.8 shows how to display group 2 of a WF/PC-1 image.

6. There are a few choices for display servers, including SAOimage, SAOtng (the next generation of SAOimage), and Ximtool. The most popular choice at present is SAOimage, and it will be described here. SAOtng may be retrieved via anonymous FTP from `sao-ftp.harvard.edu` in the directory `~ftp/pub/rd`. Ximtool may be retrieved via anonymous FTP from `iraf.noao.edu` in the directory `~pub/v2103-beta`.

Figure 2.8: Displaying an Image



You can print a hardcopy of the displayed image using the SAOimage command buttons:

1. Click “etc”.
2. Click “print”.

Modifying the Display

There are two ways to change the way in which your image is displayed:

- SAOimage command buttons can change zooming, panning, etc.
- Image intensity can be changed by resetting **display** task parameters.

Once the image appears in your SAOimage window, you can use the SAOimage commands displayed near the top third of the image window to manipulate or print your image. These commands are described in the *SAOimage Users Guide*, although most are fairly intuitive (on-line help is also available at the system level, type `man saoimage` if you are using a Unix system or `help saoimage` if you are using VMS). Click on buttons to scale the image, pan, print, or perform other commonly-used functions.

The example in Figure 2.8 shows how you should display the image for a first look. By default, **display** will automatically scale (autoscale) the image intensity using a small sampling area in the center of the image. During your first look, you may want to experiment with the image intensity scaling to improve the look of the image. This can be done using the `zscale`, `zrange`, `z1` and `z2` parameters. `Zscale` is a switch that turns the autoscaling off and on; by also setting `zscale-` and `zrange+` you can use the minimum and maximum values in the entire image as the minimum and maximum intensity values.

To choose your own minimum and maximum intensity display values, set `zscale-`, `zrange-`, `z1` to the minimum value and `z2` to the maximum value that you want displayed. For example:

```
im> disp w0mw0507v.c0h zrange- zscale- z1=2.78 z2=15.27
```

Notice in Figure 2.8 that when you run **display**, the task shows you the `z1` and `z2` values that it calculated. You can use these starting points in estimating your own reasonable values for minimum and maximum intensity values.[7]

Working with Image Sections and Groups

You display a specific group of an image by specifying the group number in square brackets following the file name.

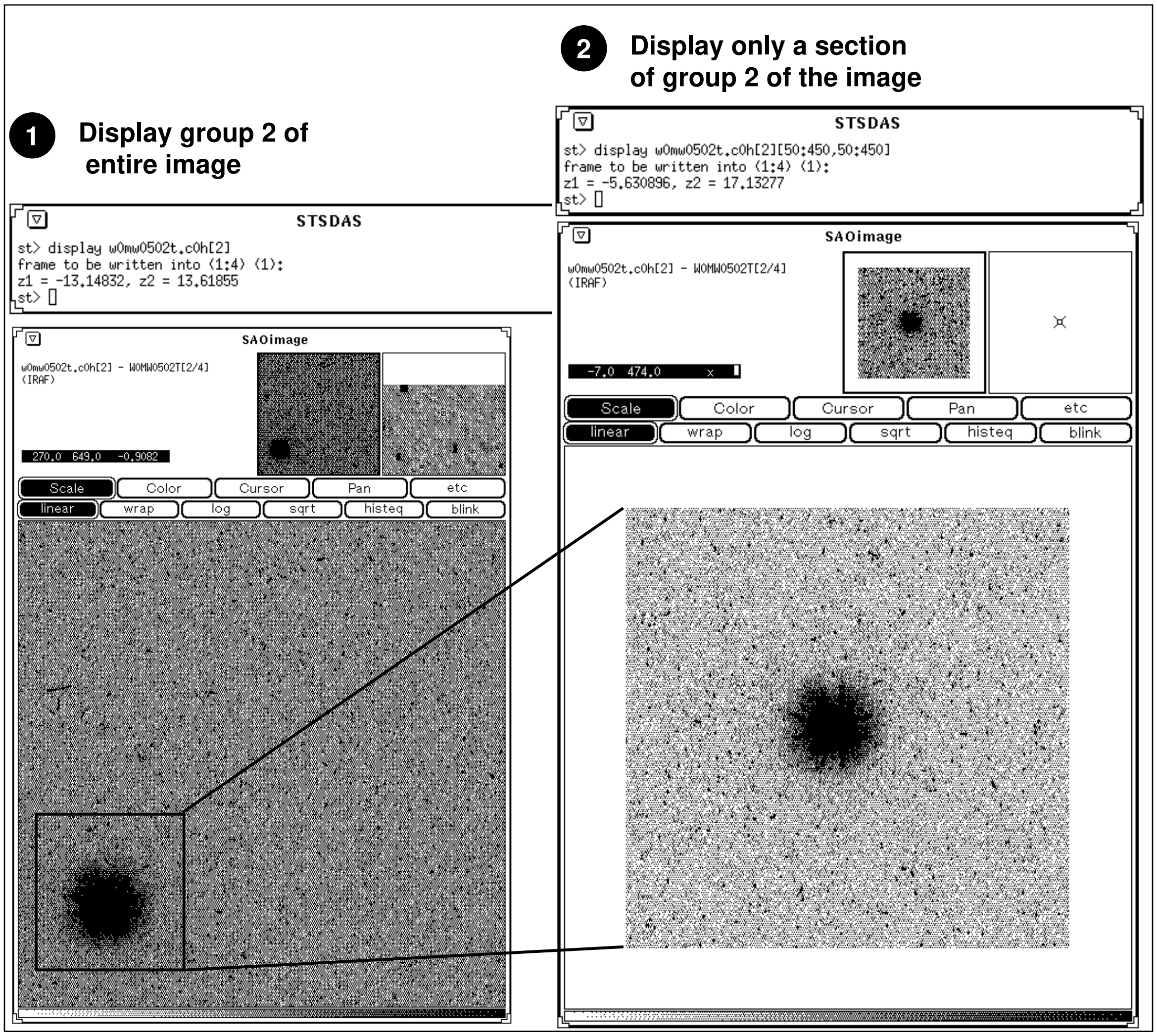
You can also work with only part of the total image by specifying a pixel range, also enclosed in square brackets. The range is the starting point and ending point, with a colon separating the two. The x (horizontal) axis is specified first, then the y (vertical) axis, separated by a comma. For example, to specify a pixel range from 101 to 200 in the x direction and all pixels in the y direction from group 3 of an image, you would use a command such as:

```
tv> display image.hhh[3][101:200,*] 1
```

If you use both group and image section syntax together, the group number must come first.

7. More information about these parameters is provided from within IRAF by typing `help display`.

Figure 2.9: Displaying Sections and Groups of an Image



Mosaic WF/PC-1 Images

WF/PC-1 images have four groups: one for each detector. You may wish to see the four images combined into one larger image. (A sample mosaic WF/PC-1 image is shown in Figure 2.10.) The **qwmosaic** task in the STSDAS **hst_calib.wfpc** package will quickly combine the four groups of a WF/PC-1 image into a single *mosaic* image that shows each group in one of the four quadrants of the image with roughly the correct orientation.

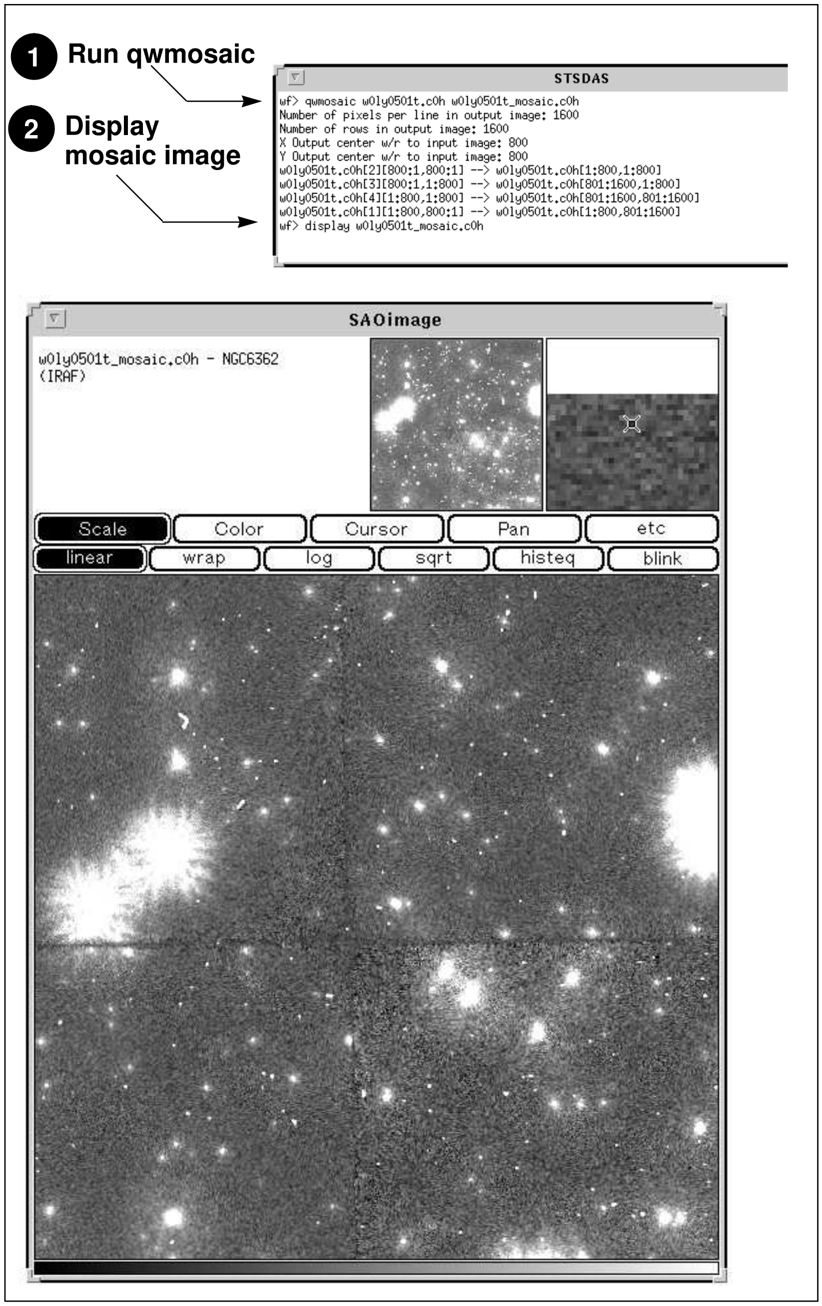
Note that **qwmosaic** is a quick version of **wmosaic** and is recommended for a quick first look at the data in mosaic form. If you need precise astrometric alignment, use the **wmosaic** task.

To produce a quick mosaic image of the four WF/PC-1 groups:

1. Run the **qwmosaic** task to combine the groups into a separate image with all groups combined.

2. Display the image using **display** (as described on page 29).

Figure 2.10: WF/PC-1 Mosaic Image



Analyzing HST Images

This section describes methods for using STSDAS and IRAF to work with two-dimensional image data from HST. Subjects include:

- A discussion of RA and Dec and how to work with coordinate information.
- Using the photometric keywords.
- Removing cosmic rays and image defects.
- A discussion of various image plotting and manipulation tasks.

In Table 2.3 through 2.6 we provide a list with brief descriptions of STSDAS and IRAF tasks you may find useful for working with your HST images. Many of these tasks are described in more detail in the sections below.

Table 2.3: Image Arithmetic and Transformation Tasks

Task	Package	Purpose
crrej	stsdas.hst_calib.wfpc	Combine images to make an image free of cosmic rays[a]
gcombine	stsdas.toolbox.imgtools	Combine images using various algorithms and rejection schemes
geomap	images	Compute a coordinate transformation
geotran	images	Resample an image based on geomap output
imcalc	stsdas.toolbox.imgtools	Perform general arithmetic on images[a]
imexpr	images	General image arithmetic.
magnify	images	Magnify an image
rotate	images	Rotate an image
wmosiac	stsdas.hst_calib.wfpc	Mosaic four WF/PC-1 or WFPC2 frames into a single image[a]

a. Handles GEIS multigroup images.

Table 2.4: Image Coordinate Tasks

Task	Package	Purpose
metric	stsdas.hst_calib.wfpc	Translate WF/PC-1 and WFPC2 pixel coords. to RA and Dec (with geo. corr.)[a]
pixcoord	stsdas.hst_calib.wfpc	Compute pixel coordinates of stars in an image[a]
rimcursor	lists	Determine RA and Dec of a pixel in an image
wcscoords	xray.xspatial	Use WCS to convert between IRAF coordinate systems
xy2rd	stsdas.toolbox.imgtools	Translate two-dimensional image pixel coordinate to RA and Dec

a. Handles GEIS multigroup images.

Table 2.5: Image Display and Graphics Tasks

Task	Package	Purpose
compass	stsdas.graphics.sdisplay	Draw north and east arrows on image display or into image itself
display	tv.display	Display an image
disconlab	stsdas.graphics.sdisplay	Display image and optionally overlay contours and coordinate grid
fieldplot	stsdas.graphics.stplot	Graph vector field
imexamine	images.tv	Examine images using display, plots, and text
implot	plot	Plot lines and columns of images
newcont	stsdas.graphics.stplot	Draw contours of two-dimensional data
north	stsdas.hst_calib.ctools	Display orientation of image based on header keywords
sgraph	stsdas.graphics.stplot	Plot spectra and image lines, overplotting error bars
siaper	stsdas.graphics.stplot	Plot science instrument apertures of HST
wcslab	stsdas.graphics.stplot	Produce sky projection grids and labels for images

Table 2.6: Image Manipulation and Utility Tasks

Task	Package	Purpose
boxcar	image	Boxcar smooth a list of images
gcopy	stsdas.toolbox.imgtools	Copy multigroup images[a]
grlist	stsdas.graphics.stplot	List file names (with groups) for all groups in image (make lists)
gstatistics	toolbox.imgtools	Computer image statistics[a]
imedit	images.tv	Fill in regions of an image by interpolation for nearby regions
noisemodel	stsdas.hst_calib.wfpc	Determine noise model parameters from CCD frames[a]
plcreate	xray.ximages	Create a pixel list from a region file (e.g., from SAOimage)
saodump	stsdas.graphics.sdisplay	Make image and colormap files from SAOimage display

a. Handles GEIS multigroup images.

RA and Dec

This section describes how to determine the orientation of your HST images and how to determine the RA and Dec location of a pixel or source on these images. Included in this section are:

- An overview of the positional information for HST images.
- Using the **rimcursor** task to determine RA and Dec.
- Using the **metric** task to determine RA and Dec, taking into consideration geometric distortion.

- Improving the astrometric zero-point accuracy.

Overview of Positional Information

Every calibrated WF/PC-1, WFPC2, and FOC image has an astrometric plate solution written in the standard FITS astrometry header keywords (CRPIX1, CRPIX2, CRVAL1, CRVAL2, and the CD matrix, CD1_1, CD1_2, CD2_1, and CD2_2). Tasks within IRAF or other packages can use this information to convert between pixel coordinates and RA and Dec. Table 2.7 lists some tasks in IRAF and STSDAS for working with positional information.

Table 2.7: IRAF and STSDAS Tasks for Working with Positions

Task	Purpose
compass	Plot north and east arrows on images
disconlab	Display image and overlay contours and coordinate grid
metric	Translate WF/PC-1 pixel coordinates to RA and Dec with geometric corrections
north	Display the orientation of an image based on keywords
rimcursor	Determine RA and Dec of a pixel in an image
wcscoords	Use WCS to convert between IRAF coordinate systems
wcslab	Produce sky projection grids for images
xy2rd	Translate pixel coordinates to RA and Dec

To find the RA and Dec of a pixel or source on an FOC image, you can use the **rimcursor** task directly on the `.c1h` file.[8] To find the RA and Dec of a pixel or source on a WF/PC-1 image, you can either use the **rimcursor** task on the WF/PC-1 mosaic image, or you can use the **metric** task directly on each individual WF/PC-1 chip image (i.e., on the individual groups). Instructions for using **rimcursor** and **metric** are described below.

Do not use the **rimcursor** or **xyeq** tasks directly on the images in the WF/PC-1 `.c0h` file if you require accurate relative positions. Calibrated WF/PC-1 images retain a residual distortion which will affect the accuracy of relative positions. Both **wmosaic** and **metric** correct for this distortion.

8. See “Limitations of the Calibration Process” on page 157 for information about calibrated FOC images.

Using rimcursor

The **rimcursor** task in the IRAF **lists** package can be used to determine the RA and Dec of a pixel on any image. The **rimcursor** task uses the astrometric header keywords to derive the coordinates of the *cursor position*. Cursor positions can be read off the image display (this is the default mode), or a file can be specified containing the (x,y) pixel positions. If you want to provide a text file of positions, set the value of the `cursor` parameter to the name of the text file. The text file must have the following format:

```
441. 410. 101 \040
208. 506. 101 \040
378.5 68.5 101 \040
```

Note that the last column can contain a letter

In this example, the 441 and 410 on the first line is the pixel location whose RA and Dec you wish to know. The "101 \040" after the pixel coordinates is formatting information is expected by **rimcursor** (see the online help for details).

To get RA and Dec in hh:mm:ss.dd and dd:mm:ss.d format, it is necessary to specify, for example:

```
cl> rimcursor w0sn0307t.c0h wcs=world \
>>>  wxformat=%12.2H wyformat=%12.1h
```

The number after the decimal in the format specifications is the number of digits to print after the decimal in the seconds (of time or arc). The 12 is the field width. Capital "H" means divide the number by 15 (to convert from degrees to hours) before formatting, while "h" just prints in dd:mm:ss.d format.

The specified pixel coordinates in this file can be fractional pixel values, perhaps derived through the use of a source centroiding task such as **imcntr** (found in the **proto** package) or from **daophot** (in the **digiphot** package).

Using metric for WF/PC-1 and WFPC2 Images

The **metric** task in the STSDAS **hst_calib.wfpc** package determines the RA and Dec of any pixel (from any group) in a WF/PC-1 or WFPC2 multigroup GEIS file, taking into account (and correcting for) the geometric distortions of the specified CCD chip and its offset relative to the reference chip. The reference chip is always at group 2, i.e., for the Wide-Field Camera in WF/PC-1 it is WF2 and for the Planetary Camera it is PC6. After applying the geometric and transformation corrections, **metric** uses the reference chip's group parameters CRVAL1, CRVAL2, CRPIX1, and CRPIX2 in conjunction with the CD matrix coefficients to translate the corrected pixel coordinates to RA and Dec. The resulting RA and Dec will therefore be in the epoch of these group parameters, usually J2000.

The **metric** task can take three kinds of inputs:

- A single pair of (x,y) coordinates, specified by using the `x` and `y` input parameters.
- A set of x and y pixels passed in a table, specified by passing the table and column name (separated by a space) for the `x` and `y` parameters.

- Interactive cursor input, specified by setting `x` to "", which will cause the image to be displayed. You can then move the cursor to the desired position and press a key to get the coordinates. Pressing Q will end the task; pressing C will enable centroiding; pressing any other key will return the x,y coordinate at the cursor position.

Centroiding is used to calculate a refined center position of a point source in the image by using the given position as an approximate initial value. Choosing a proper box size for centroiding is important: for example, if there is another star close to the target star, a large box will make the centroid lie between these two stars, instead of at the center of the target star. The algorithm used here is identical to the IRAF task **imcntr**. (Refer to the help file of **imcntr** for details of the algorithm.)

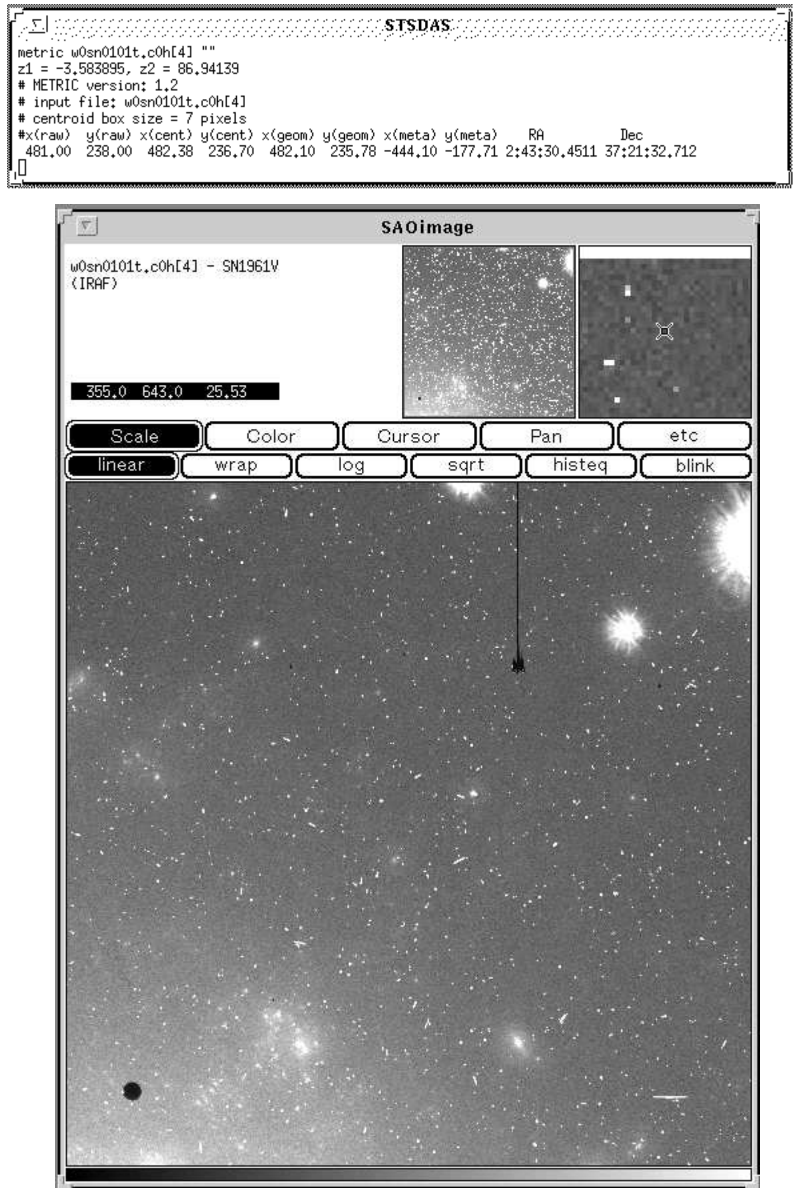
Results are displayed on your terminal screen. The output will contain the following columns:

- Original input pixel coordinates.
- Pixel coordinates after centroiding (if applicable).
- Coordinates after geometric corrections.
- Coordinates after transformation to the reference chip.
- RA and Dec.

Figure 2.11 shows an example of **metric** being use to determine the centroided location of a source in a WF/PC-1 group 4 image via the interactive cursor input. Before running **metric** in this mode, you must start SAOimage (or another display server). See the discussion of image display on page 29.

The **metric** task cannot be used on WF/PC-1 or WFPC2 mosaicked images. Use the **rimcursor** task directly on these images.

Figure 2.11: Determining Centroided Star Position Using metric



Improving Your Astrometric Accuracy

Differential astrometry (measuring a position of one object relative to another in an image) is easy and relatively accurate for HST images, while absolute astrometry is more difficult due to uncertainties in the locations of the instrument apertures relative to the Optical Telescope Assembly (OTA or V1) axis and the inherent uncertainty in the Guide Star positions. However, if you can determine the position of any single star in your HST image, then your absolute astrometric accuracy will be limited only by the accuracy with which you know that star's location and image orientation.

If there is a star on your image suitable for astrometry, you may wish to extract an image of the sky around this star from the Digitized Palomar Sky Survey and measure the position of that star using, for example, the GASP software. This can provide an absolute positional accuracy of approximately 0″.7. Contact the Help Desk for assistance (send e-mail to `help@stsci.edu`).

Photometry

Included in this section are:

- A description of how to use the header keyword information to convert from counts to flux or magnitude.
- A description of some of the tasks within IRAF and STSDAS that may be useful for determining source fluxes.
- A description of how to use the STSDAS **synphot** package to re-derive your flux scale.

Converting Counts to Flux or Magnitudes

All calibrated HST images have units of counts or Data Numbers (DN). During the correction for absolute sensitivity in pipeline calibration, **calfoc**, **calwfp**, and **calwp2** do not alter the units of the pixels in the image. Instead they calculate and write the conversion factors, the inverse sensitivity PHOTFLAM, and the zero point of the magnitude scale, PHOTZPT, into header keywords in the calibrated data. For WF/PC-1 and WFPC2 images, these values are written as group parameters (i.e., they are assigned for each group). To convert from data numbers (or counts) to flux in units of erg/cm^2/s/Å, multiply by the value of the PHOTFLAM header keyword and divide by the value of the EXPTIME keyword (exposure time). You can use the STSDAS task **imcalc** to convert an entire image from counts to flux units.

You can use **hedit** on an FOC image to determine the values of the PHOTFLAM and EXPTIME header keywords and then convert the image to flux units using **imexpr**. For example:

```
st> imexpr "a*b/c" outimage.hhh \
>>>  a=x0fp0102t.c1h b=a.photflam \
>>>  c=a.exptime
```

If, instead, you wish to convert to magnitudes, magnitudes are computed as:

$$m = -2.5 \times \log 10\ (F) + \mathrm{PHOTZPT}$$

where:

- PHOTZPT is the zero-point of the ST magnitude scale as given by the value of the PHOTZPT header keyword.
- F is the flux = counts x PHOTFLAM / EXPTIME.

The zero point of the ST magnitude system is –21.10. This value is chosen so that Vega has an ST magnitude of zero for the Johnson V passband (see Koornneef et al., 1986; Horne, 1988; and the *Synphot Users Guide*).

The inverse sensitivity (as given by the PHOTFLAM) keyword is the key to converting from counts or DN to flux units. PHOTFLAM is defined to be the *mean* flux density (i.e., the flux density of a source with a spectrum which is flat in f_λ across the bandpass of your observation) that produces a count rate of 1 per second with the HST observing mode (PHOTMODE) used in your observation. If the spectrum of your source is significantly different from flat (e.g., if you have observed a source dominated by an emission line or with a strong slope across the bandpass), you may wish to recalculate the conversion factor from counts to flux units for your particular source, using **synphot** (below). In addition, WF/PC-1 observers should note that the PHOTFLAM value calculated by **calwfp** does not include a correction for temporal variations in the throughput of the instrument due to contamination buildup. For WF/PC-1 observations of sources blueward of 5000 Å, a correction should be applied to the flux calibration of your images for this effect. Likewise, FOC observers should note that the PHOTFLAM value calculated by **calfoc** does not correct for format differences in sensitivities; FOC observations made in formats other than 512 x 512 should be corrected for this effect (see Table 8.4: "Format-Dependent Sensitivity Ratios." on page 166).

If your HST image contains a source whose flux you know from ground based measurements, you may choose to check or determine the final photometry of your HST image from the counts observed for this source.

Using Synphot to Re-derive your Flux Scale

The conversion of observed count rates to absolute fluxes (i.e., the process of flux calibration) is accomplished for HST imaging detectors by using both knowledge of the instrumental sensitivities and principles of synthetic photometry. In this section we introduce the theories and principles (below) and then provide a section of examples of how to actually use **synphot** (page 44).

Theory

The detected count rate through a broad passband P is given by:

$$C(P) = \frac{A}{hc} \int P(\lambda) \; f_\lambda(\lambda) \, \lambda \; d\lambda$$

where:

- A – is the nominal collecting area of the telescope.
- $f_\lambda(l)$ – is the flux density as a function of wavelength of the target.
- $P(l)$ – is the dimensionless bandpass throughput function.

- the division by $h\nu = hc/l$ converts the energy flux to a photon flux, as is appropriate for photon-counting detectors. Alternatively, we can write this expression as:

$$C(P) = \frac{f_\lambda(P)}{U_\lambda(P)}$$

where:

- $f_\lambda(P)$ – denotes the appropriate average of the star's flux density spectrum over the passband P.
- $U_\lambda(P)$ – is the flux density required to produce a unit response of 1 count s^{-1} within the passband and is referred to as the *inverse sensitivity* of the photometric mode.

Following this formulation, the precise definition of mean flux density in a broad passband must be:

$$f_\lambda(P) = \frac{\int P(\lambda) f_\lambda(\lambda)\,\lambda\,d\lambda}{\int P(\lambda)\,\lambda\,d\lambda}$$

and the count rate to flux density conversion factor for a broad passband is:

$$U_\lambda(P) = \frac{hc}{A\int P(\lambda)\,\lambda\,d\lambda}$$

So for a given observed count rate C, the corresponding mean flux density within the passband is:

$$f_\lambda(P) = U_\lambda(P)\,C$$

Corresponding magnitudes are simply given by:

$$m_\lambda(P) = -2.5\,\log\,U_\lambda(P)\,C + K$$

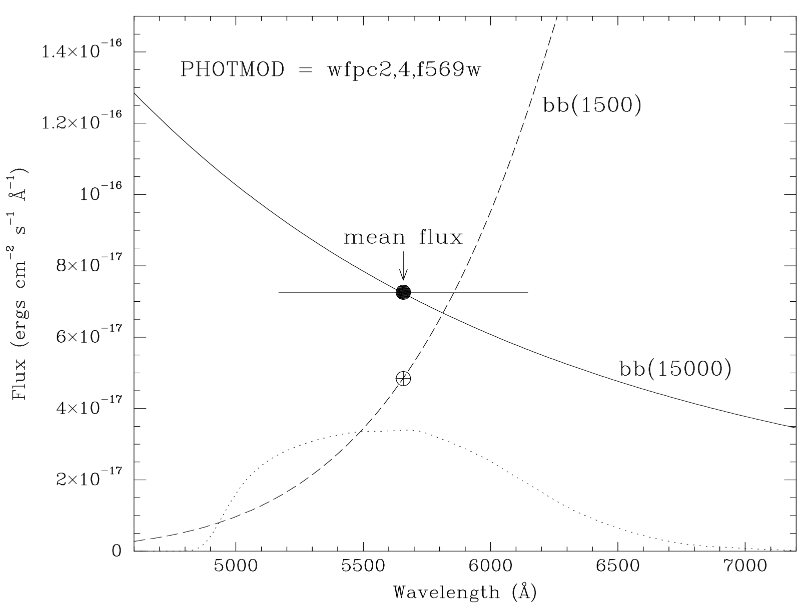
where K is chosen for convenience to be –21.10 (for U_λ in units of erg cm^{-2} $Å^{-1}$) so that m_λ (5500 Å) is approximately equal to the Johnson V magnitude.

Notice that the mean flux density depends on the spectrum and the passband shape but not on its overall normalization, while the conversion factor U depends only on the integrated area of the passband—it does ***not*** depend on the spectral shape of the source being observed. The value of U is computed during the routine pipeline processing of WF/PC-1 and FOC imaging observations based on the known throughput function of the observing mode in use and is recorded in the image header keyword PHOTFLAM. To compute the mean absolute flux density of a source within the observed passband one merely has to multiply

PHOTFLAM by the observed countrate of the source. This process yields the correct value for the *mean* flux density of the source as multiplied by the passband in use, no matter what the spectral characteristics of the source may be. This does *not* necessarily yield the intrinsic flux density of the source at the effective wavelength of the passband. Only in the case of a source with a spectrum that has constant f_λ as a function of wavelength will this calibration process yield the intrinsic flux density of the source at any wavelength within the passband.

For example, Figure 2.12 shows flux spectra for two blackbody sources, with temperatures of 1500 K and 15000 K over the wavelength range of the WFPC2 F569W passband (PHOTMOD). Both spectra have been normalized so that they produce an observed count rate of 100 counts per second in the passband. For this passband PHOTFLAM=7.26E-19 erg/cm^2/Å/count, hence both of these spectra have a *mean* flux density of 7.26E-17 erg/s/cm^2/Å within this passband. In the case of the 15000 K source, the mean flux density also happens to be nearly identical to the actual flux density at the effective wavelength (or *pivot wavelength* in **synphot** terminology) of the passband (~5657 Å). As you can see from the figure, the flux density of the 1500 K source at the pivot wavelength of the passband is significantly lower (~4.8E-17 ergs/s/cm^2/Å). In Figure 2.12, the dotted line shows the shape of the passband throughput function. The mean flux within the passband is indicated by the filled circle and the horizontal line on either side shows the equivalent FWHM extent of the passband. The open circle indicates the flux density of the 1500 K spectrum at the pivot wavelength of the passband.

Figure 2.12: Spectra of a 1500 K and a 15000 K Blackbody, Normalized to Produce 100 counts/sec in the WFPC-2 F569W Observing Mode



Using Synphot

If the spectral type of a source is known, the intrinsic flux density spectrum can be computed using tasks from the STSDAS synthetic photometry (**synphot**) package. This is done by using the **synphot** tasks to create a spectrum that produces the same observed countrate within a given passband as what was obtained in an actual WF/PC-1, WFPC2, or FOC image. If the observed source is known to have a stellar-like spectrum, a matching spectral type can be found in one of several spectral atlases that are available for online use at STScI and can be exported to off-site locations (see page 93 of the *Synphot User's Guide* for more details on the availability and contents of these atlases). Alternatively, one can create simple blackbody and powerlaw spectra using built-in **synphot** spectral synthesis capabilities, or you can use your own independent spectral data for the same source or similar sources.

For example, let's say you have an FOC observation of a source that is known to have a G5V spectral type and the observation used the f/96 relay and the f430w filter. Let's also say that the source produced an observed countrate of 152.3 counts s^{-1} in the image. There is a spectrum of a G5V star in table number 51 of the Bruzual-Persson-Gunn-Stryker spectral atlas. You can use the **synphot calcspec** task to create a G5V spectrum that produces this same countrate as shown below.

Figure 2.13: Creating G5V Spectrum with calcspec

```
sy>calcspec "rn(crgridbpgs$bpgs_51,band(foc,f/96,f430w),152.3,counts)" \
>>> g5spec.tab flam
```

Here we are using the renormalize, "`rn`", function to produce a spectrum that has the desired observed countrate. The renormalize function takes four arguments: the input spectrum, which in this case we are reading from the table `crgridbpgs$bpgs_51.tab`; the passband over which to compute the normalization, which in this example is defined by the FOC,f/96,F430W observing mode; the desired normalization value, 152.3; and finally the units of the normalization value, counts. The computed spectrum will be placed in the output table `g5spec.tab` and the units of the output spectral data are specified to be in units of f_λ by the parameter `flam`. The resulting spectrum can be plotted using the **synphot** task **plspec**. From this you can determine the spectral flux density at any desired wavelength.

The format of some input to **synphot** tasks was changed with the introduction of a new expression evaluator in November 1993. You may need to update your software or see the *Synphot User's Guide* to find the equivalent syntax for the older versions.

Other synthetic photometry operations can also be performed on this spectrum, such as determining what the flux density or magnitude of this source would be in the standard Johnson-Cousins *UBVRI* passbands or in other HST passbands. This can be accomplished using the **calcphot** task as follows:

```
sy> calcphot flam v g5spec.tab
```

which will calculate the mean flux density of the spectrum within the Johnson *V* passband, which turns out to be 4.33 x 10^{-16} ergs/s/cm^2/Å, or:

```
sy> calcphot vegamag v g5spec.tab
```

which will calculate the *V* magnitude (relative to Vega) of the spectrum (which is 17.30), or:

```
sy> calcphot counts wfpc2,4,F439W g5spec.tab
```

to determine what countrate this source will produce in the WFPC2 F439W passband for detector WF4 (80 counts per second).

IRAF and STSDAS Photometry Tasks

In Table 2.8 we list some of the tasks available within IRAF and STSDAS for determining source (or background) counts and magnitudes.[9]

Table 2.8: Tasks and Packages for Determining Counts and Magnitudes

Task or Package	Purpose
daophot and apphot	Tasks for doing stellar photometry, see online help files for details. daophot contains tasks for doing crowded-field photometry, apphot for doing aperture photometry. Both are in the **digiphot** package.
imstat	Compute and print image pixel statistics (in **images** package).
imcnts	Sum counts over specified region, subtracting background. Can be used with mask file (such as from plcreate). In **xray.xspatial** package.
wstat	Compute and print image pixel statistics (in STSDAS **hst_calib.wfpc**).
plcreate	Create a pixel mask from a region descriptor (in **xray.ximages**).
isophote	Tasks for fitting and graphing elliptical isophotes (in STSDAS **analysis**) package. See online help for details.

The **apphot** package provides an option for measuring flux within a series of concentric apertures—this can be used to determine the flux in the wings of the PSF, which is useful if you wish to estimate the flux of a saturated star by scaling the flux in the wings of the PSF of the saturated star by the ratio of total to wing flux for unsaturated stars.

9. A general discussion of doing photometry in IRAF is described in the document "Photometry Using IRAF" by Lisa A. Wells, February 1994. Available from NOAO via anonymous FTP to `tucana.noao.edu`. See also "A User's Guide to Stellar CCD Photometry with IRAF," by P. Massey and L. Davis, 1992, also available from NOAO.

Removing Cosmic Rays and Image Defects

There are many tasks in IRAF and STSDAS that can be used to remove cosmic rays and other image defects from your HST images. In Table 2.9, we list some of these tasks and briefly explain their functions. Of particular interest to HST observers are:

- The **crrej**, **gcombine**, and **cosmicrays** tasks to remove cosmic rays from WF/PC-1 or WFPC2 images.
- The **imedit** task to remove image defects. This is particularly useful for removing reseau marks from FOC images.

These are described in more detail below.

FOC images do not suffer from cosmic rays.

Table 2.9: Tasks for Removing Image Defects

Task	Function
cosmicrays	Detect and replace cosmic rays in a single image, based on flux of each pixel relative to mean flux of its neighbors—works on multigroup images (in **noao.imred.ccdred** package).
gcombine	Combine a set of images to produce a weighted average or median, pixels may be rejected using various algorithms (in **stsdas.toolbox.imgtools** package).
crrej	Generate an image free of cosmic rays from multiple exposures of the same field—works with group format images (in STSDAS **hst_calib.wfpc** package).
imedit	General utility for examining and editing pixels in an image (in the **images.tv** package).
rremovex	Remove reseau marks from FOC images by filling pixels under reseau marks with average of neighboring pixels (in the STSDAS **hst_calib.foc.focphot** package).

Removing Cosmic Rays

If you have multiple images of the same field, you can use the **crrej** task to remove cosmic rays and image defects. If you have only a single image of a given field, you can use the **cosmicrays** task to detect cosmic rays and replace them with the average value of the surrounding pixels.

Using the cosmicrays Task

The **cosmicrays** task (in **noao.imred.ccdred**) detects potential cosmic rays (or more generally, potential bad pixels) by searching for the brightest pixel which is greater by an amount defined by the `threshold` parameter than the average value of the pixels (excluding the two brightest pixels) in a given area, where the size of the area is defined by the `window` parameter. The task then subtracts a background plane from the pixels in the window (determined by fitting to the edge pixels of the window). The potential cosmic ray pixel is now declared a *bad pixel* if its background subtracted value divided by the mean (background subtracted) flux in the window is greater than the value of the `fluxratio` parameter. The **cosmicrays** task records the mean background subtracted flux in the window and the ratio of the bad pixel and background fluxes. The task only considers the brightest pixel in a given window as a potential bad pixel in a given pass, however you can specify the number of detection passes the task should perform by setting the `npasses` parameter.

A plot can be made of the flux ratio as a function of the mean flux and an interactive mode is available (by setting the parameter `interactive=yes`) which allows you to examine the pixels flagged by the task, alter the `threshold` and `fluxratio` levels, delete or undelete pixels flagged by the task, and display a surface plot of the window around any candidate. The **cosmicrays** task can create a bad pixel file (specified by the `badpix` parameter) which contains a list of the coordinates of the bad pixels found by the task. You may also wish to examine the image statistics (for example, using **gstatistics**) of your pre- and post-corrected image to assure that you have not flagged real features. The **cosmicrays** task works on only a single group of a multi-group file at a time.

Using crrej

The **crrej** task (in the STSDAS **hst_calib.wfpc** package) is designed to take multiple exposures of a given field and combine the images while rejecting very high counts in each pixel stack. It is a general purpose task, but was designed with WF/PC-1 images in mind. The task begins with a *guess image* (usually either the minimum or median of the input images, as specified through the `initial` parameter). Cosmic ray pixels are selected as those pixels with values greater than or less than *N* times the noise (where *N* is specified through the `sigmas` parameter). The *noise* is determined using the **noisepar** parameter set. The **noisepar** parameters specify the coefficients of the model used to estimate the noise. The total noise is the square-root of the quadratic sum of the pixel DN value (scaled by the detector gain specified by the `gain` parameter), the CCD readout noise (specified by the `readnoise` parameter), and a term that scales as a percentage of the DN value (as given by the `scalenoise` parameter). Stack pixels which are *N* times the noise bigger or smaller than the guess image value are considered *bad* and are not used in creation of the output, cosmic-ray free, average image. It is possible to automatically discard pixels adjacent to rejected pixels (by setting the `radius` parameter) or to reject adjacent pixels based on their own rejection threshold (the `pfactor` parameter). You can also disable the discarding of neighboring pixels for rejected pixels greater than a certain value (as specified through the `hotthresh` parameter) This is useful for dealing with hot pixels, which do not affect neighboring pixels as do cosmic rays. You may wish to perform the rejection process several times to allow the solution to slowly achieve an equilibrium and to allow information about rejected pixels to propagate into adjacent pixels. The default setting `sigmas=4,4,3,2` runs the program four times with *N*=4, 4, 3, and 2, respectively, in the four iterations.[10] The **crrej** task will work on multigroup images. See the online help for more details.

The **crrej** task should only be used on a stack of images where the input images have similar exposure times, and the images are well aligned.

Using imedit

The **imedit** task in the IRAF **images.tv** package is a sophisticated interactive image editing task that allows you to select pixels for editing and to replace them with either a value derived from neighboring pixels, a value derived from a user-defined background area, or a predetermined value (to flag pixels, for example). The pixels to be edited can be selected interactively, or from a list.

Background-subtracted image statistics can be determined and displayed for subregions around the selected pixels. The image can be redisplayed after each editing of a pixel (or after some number of pixel edits). Extensive online help for **imedit** is provided, which includes detailed instructions on setting input parameters.

Removing Reseau Marks from FOC Images

One way that you might use **imedit** is to remove reseau marks from an FOC image.

If you need to remove only a few reseau marks from your image, and they are obvious, then the easiest approach is to remove them interactively. If you have an accurate list of positions for the reseau marks on your image, you can automatically remove them by feeding a cursor file to **imedit**. Lists of reseau positions are provided on STEIS. However, since the reseau marks move over time, the positions in this list will not accurately match the positions of the reseau marks in your image. You can still use these lists as inputs to **imedit**, but you will need to turn on the search option of **imedit**. Figure 2.14 shows an example of a cursor file to remove four reseau marks. Here, we have enabled the search option by specifying `search = -5`; this will search an area around the pixel location specified and find the minimum pixel in the region. Nevertheless, this method will only work if the background level in your FOC image is high enough that **imedit** will recognize the reseau marks as the local minimum.

10. Note that the task assumes that the images in the stack are registered (which will be the case with WF/PC-1 CR-SPLIT images). If your images are not registered, you will need to first register them before using **crrej**.

Figure 2.14: Sample imedit Cursor File for Removing FOC Reseau Marks

```
# Set parameters that are different from the defaults.
:aperture square
:search -5.
:radius 3.
:buffer 5.
:width 5.
:value 0.
:sigma INDEF
:xorder 2
:yorder 2
# These are the X and Y pixel coordinates of four reseau marks
# to be removed.  The "b" at the end of the line is the cursor key
# that one would use in interactive mode.
248 310 1 b
313 317 1 b
243 369 1 b
308 376 1 b
```

It you have a flatfield image taken before or after your science image, then you can find reseau marks in the flats using the **resfind** task (in the **hst_calib.foc.focgeom** package). This tasks works non-interactively to produce a reseau table (a STSDAS binary table with the pixel locations of the reseau marks) as output. To use this table as input to **imedit**, you will first need to convert it to an ASCII table. To do this run **rprintx** on the reseau table, setting `plain=yes` and directing the output to a file, for example:

```
fo> rprintx res.table plain=yes > reseau.cur
```

After running **rprintx**, you need to edit the file to append " 1 b" to each line. That file could then be passed as the cursor file to **imedit**. When running **imedit** on a file created in this way, turn off the search option (set `search` = `0`), since the reseau positions determined from the flats should accurately match those in your science images.

Plotting and Manipulating Image Data

This section describes two basic tools for manipulating and working with image data, such as **implot**, **imexamine**, and **contour**.

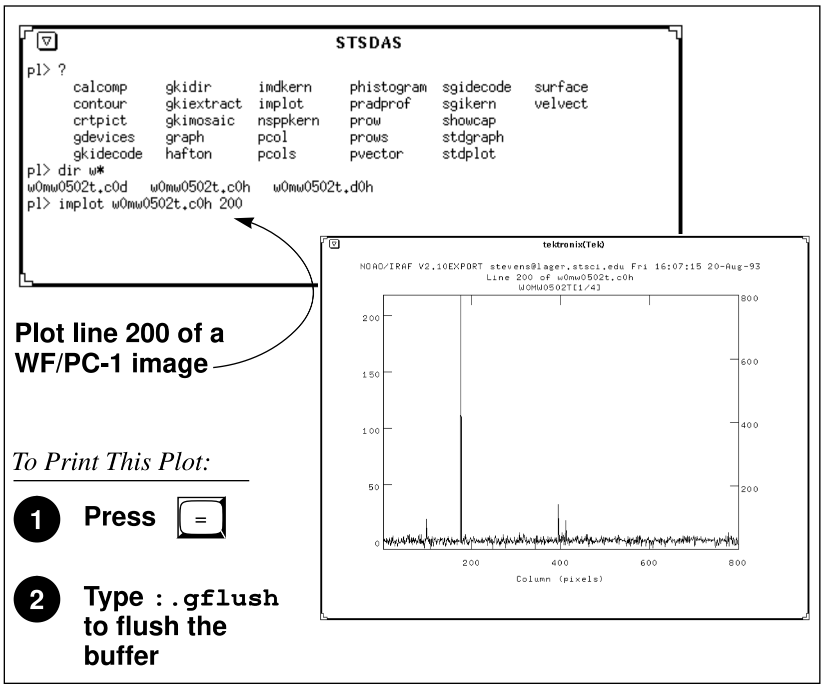
Using implot

The IRAF **implot** task (in the **plot** package) will let you interactively examine an image by plotting data along a given *line* (x axis) or *column* (y axis). When you run the task, a large number of commands are available, in addition to the usual cursor mode commands common to most plotting tasks; a complete listing of commands is found in the on-line help, but the most common are listed in Table 2.10. Figure 2.15 shows an example of how to use the **implot** task.

Table 2.10: Basic implot Commands

Keystroke	Command
?	Display on-line help
L	Plot a line
C	Plot a column
Q	Quit implot
Space	Display coordinates and pixel values

Figure 2.15: Plotting Image Data with implot



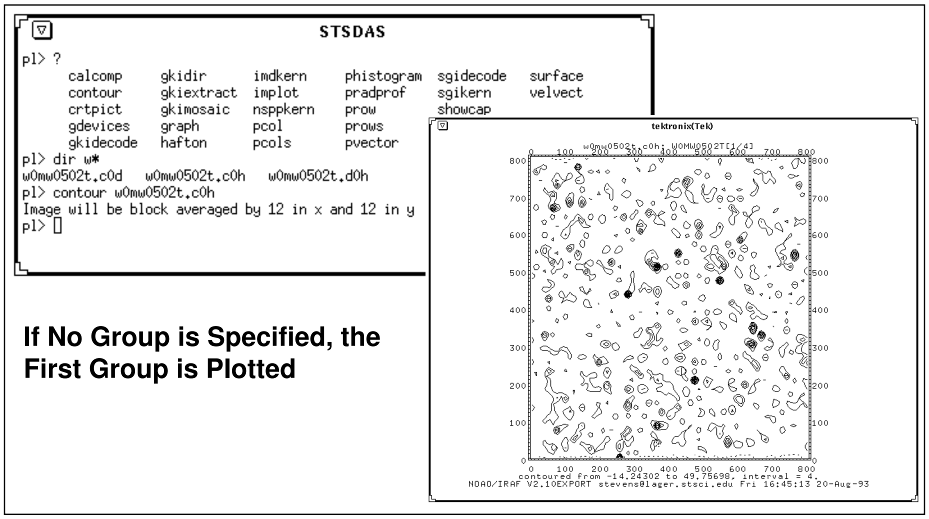
Using imexamine

The IRAF **imexamine** task (in the **images.tv** package) is a powerful task that integrates image display with various types of plotting capability, and that provides the ability to do simple photometry measurements. Commands can be passed to the task using the image display cursor and the graphics cursor. A complete description of the task and its usage are provided in the online help, available from within the IRAF environment by typing `help imexamine`.

Contour Plots

Contour plots of image data (either WF/PC-1, WFPC2, or FOC) can be created using the **contour** task in the IRAF **plot** package, or the **newcont** task in the STSDAS **graphics.stplot** package. Figure 2.16 is an example of how to use the task with a WF/PC-1 image. The **newcont** task provides much more flexibility in the specification of contour levels to be plotted, computes contours using a more advanced algorithm, and allows the perimeter to be labeled in world coordinates.

Figure 2.16: Contour Plots Using contour



Displaying HST Spectra

This section explains how to generate plots of FOS or GHRS spectra for a quick first look. For FOS and GHRS data, the final calibrated data encompasses two files: the `.c1h` file, which contains the calibrated flux values for each pixel and the `.c0h` file, which contains calibrated wavelength values for each pixel. Before we delve into the individual plotting tasks, we will begin with some general information about producing hardcopy plots and PostScript.

There are three basic plotting tasks that can be used to produce a quick-look plot of FOS or GHRS spectra:

- The **fwplot** task produces a flux vs. wavelength plot of your spectra.
- The **grspec** task plots all (or multiple) groups from a given dataset to a single plot.
- The **splot** task (in the **onedspec** package) can be used to plot and analyze simple one-dimensional spectra.

Each of these tasks will be described in more detail in subsequent sections.

Producing Hardcopy

In this section we will describe how to get hardcopy from most IRAF tasks, how to use the Interactive Graphics Interpreter (IGI) in conjunction with plots, and how to produce PostScript plots.

Hardcopy Plots from fwplot and Most IRAF Tasks

To print a copy of the displayed plot:

1. Type `=gcur` in the command window (where your CL prompt is located).

2. Move the cursor to any location in the graphics window.
3. Press [=] to write the plot to the graphics buffer.
4. Type `q` to exit graphics mode.
5. At the cl prompt, type `gflush`.

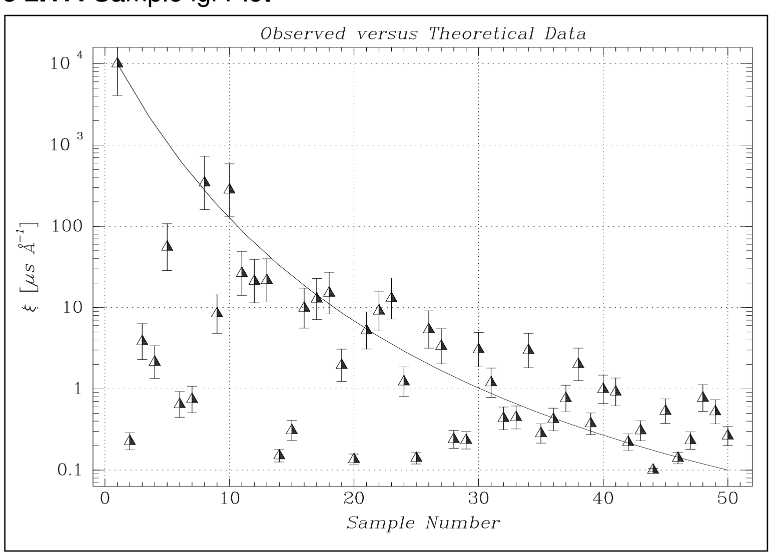
Plots will be printed on the printer defined by the environment variable `stdplot`. (See "Setting Environment Variables" on page 22).

If you wanted the plot saved to a PostScript file, for use with other applications, you could do so using the **psikern** PostScript kernel, which is described in "PostScript Plots" on page 52.

Using igi

As your plotting needs grow more sophisticated—and especially as you try preparing presentations or publication-quality plots—you should investigate the Interactive Graphics Interpreter, or **igi**. This is a task in the STSDAS **stplot** package that can be used to draw axes, error bars, labels, and a variety of other features on plots. Options are available to use a number of different line weights, font styles and feature shapes, enabling you to create complex plots. Figure 2.17 shows a sample plot created in **igi**, however, because **igi** is a complete graphics environment in itself, it is well beyond the scope of this document. You can learn more about **igi** by requesting a copy of the *IGI Reference Manual* from the Help Desk at STScI (`help@stsci.edu`).

Figure 2.17: Sample igi Plot



PostScript Plots

Using the PostScript kernel, **psikern**, you can create PostScript files from your plots that can be easily incorporated into papers, reports, or other documents created by LaTeX, FrameMaker, or other applications that understand PostScript graphics. You can also use **psikern** to add color to your plots and to use PostScript fonts.

There are two commonly-used ways to create a PostScript file of your plot (additional features are explained in the on-line help):

1. With the plot displayed, type: `:.snap psi_port` for a portrait-orientation plot (or `psi_land` for landscape orientation).
2. Before running your **plot** task, set the `device` parameter to `psi_port` or `psi_land`.

The system will automatically name the file, and usually store it on a temporary disk. You should then move the file to an appropriate directory where it won't be deleted.

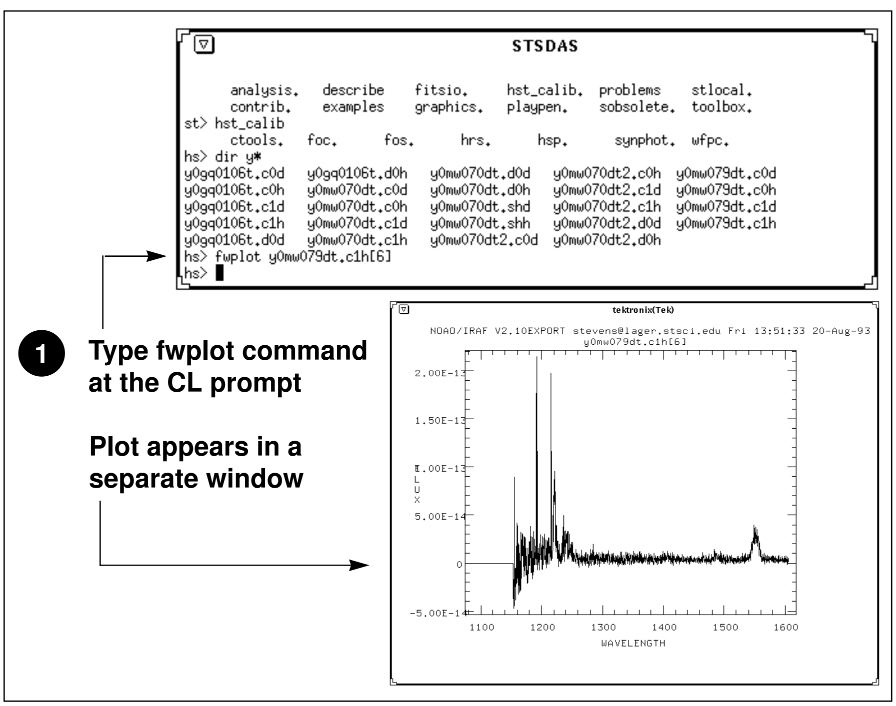
Using fwplot

GHRS and FOS observers can quickly produce a flux vs. wavelength plot of a spectrum using the **fwplot** task in the STSDAS **hst_calib.ctools** package. This takes a single argument—the name of the calibrated flux (`.c1h`) file. To use the **fwplot** task:

1. Load the **hst_calib** package.
2. Type the **fwplot** command, giving the filename and any group number that you want plotted. For example:

```
hs> fwplot y0mw070dt.c1h[6]
```

Figure 2.18: Simple Flux Versus Wavelength Plot of Spectral Data



Remember that for FOS ACCUM mode data, the last group contains the spectrum for the full exposure, while for GHRS ACCUM mode data, the groups are separate subintegrations that need to be added together to produce the final spectrum (see page "Addition With Wavelength Alignment" on page 57 and "FP-SPLITs" on page 375).

Error Bars in fwplot

The **fwplot** task can print error bars from the statistical error file along with the data. To print error bars, set the parameter `plterr=yes`. For example:

```
ct> fwplot y0gq0106t.c1h[6] plterr=yes
```

grspec

The **grspec** task can be used to plot many groups from a single image on a single plot. The task takes as input the name of the image and a list of group numbers, or range of group numbers, to plot. For example, to plot groups 2 and 5, you could use a command such as:

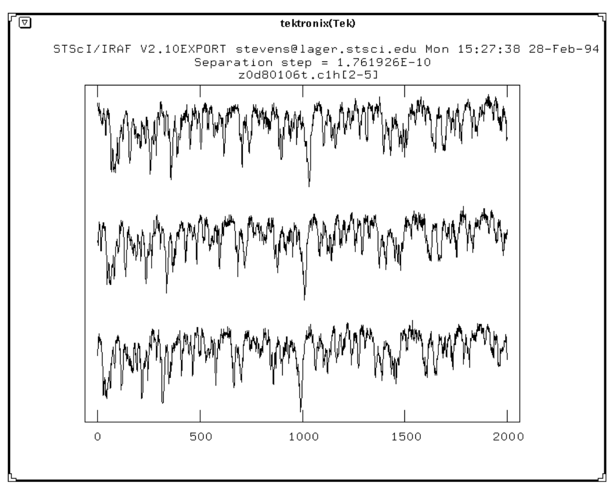
```
st> grspec y0mw070dt2.c1h 2,5
```

Note that the group number is passed as a separate parameter, *not* with the standard group syntax. You could also plot a range by using a dash instead of a comma between the group numbers, for example. Figure 2.19 shows an example of the **grspec** task being used to plot groups 2 through 5 of a GHRS spectra.

```
st> grspec z0d80106t.c1h 2-5
```

Figure 2.19: Using grspec

Note that group syntax is not used with grspec



splot

The **splot** task (in the **onedspec** package of IRAF) plots and fits one-dimensional spectral data. You can use the task directly on your calibrated HST spectra in two ways:

- Plot flux versus pixel by using the `.c1h` file.
- Plot wavelength versus pixel using the `.c0h` file.

You will probably find **splot** most useful after combining the flux and wavelength information from the `.c0h` and `.c1h` files into a single file. How to do that (using the **mkmultispec** task) and how to use **splot** to analyze your HST data is described in detail in the next section. .

Analyzing HST Spectra

This section describes some STSDAS tasks that can be used for analyzing and manipulating spectral data. Included here are descriptions of how to:

- Coadd spectra.
- Combine wavelength and flux information.
- Perform other general spectral analysis functions with STSDAS.

Coadding Spectra

There are several reasons why you may wish to coadd HST spectra, for example:

- To produce a summed spectrum from the groups of a FOS or GHRS RAPID-mode observation.
- To produce a final calibrated spectrum from GHRS observations taken in FP-SPLIT mode.
- To combine spectra from separate exposures (i.e., to coadd datasets).

In this chapter we describe two different ways to do this in STSDAS. The first method is to simply coadd the data, pixel by pixel. This is appropriate whenever there is no shift in wavelength or pixel space across the spectra you are coadding (e.g., for FOS rapid mode spectra or some GHRS rapid mode spectra)[11]. The second is to either use the wavelength information in the `.c0h` file or to cross-correlate spectral features in the spectra themselves (`.c1h` file) to align the data in wavelength space prior to coaddition. This method is appropriate for

---

11. Because there is no onboard Doppler compensation applied to GHRS rapid mode observations, for high resolution GHRS rapid mode spectra, you may wish to use the **poffsets** method to remove Doppler shifts due to spacecraft motion prior to coadding—see page 58.

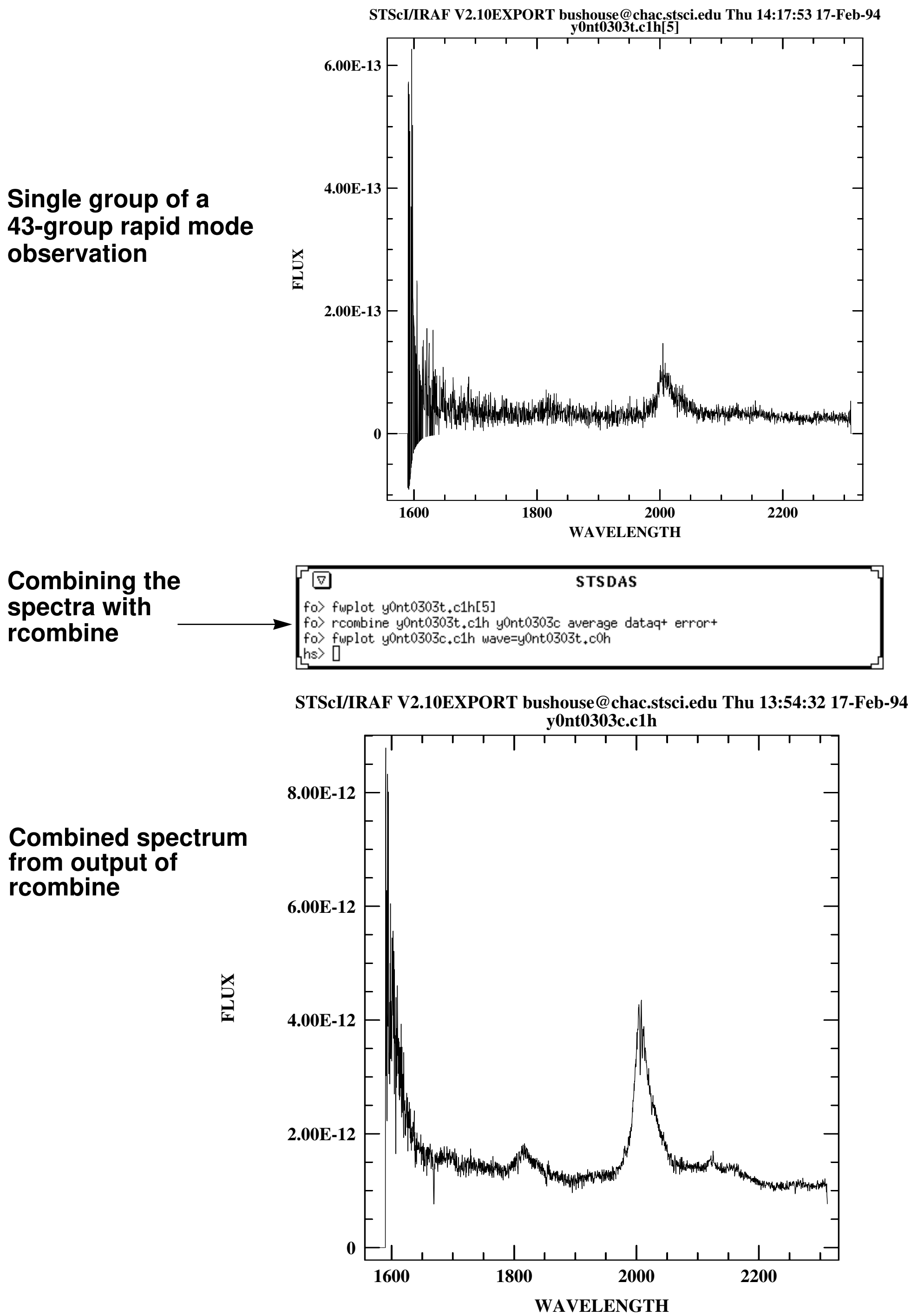
GHRS ACCUM mode spectra taken with FP-SPLIT, or when coadding individual datasets when the filter grating wheel has been repositioned between the two observations (causing the potential for a shift in wavelength space).

rcombine

The **rcombine** task (in the STSDAS **hst_calib.ctools** package) coadds or averages, pixel-by-pixel, the groups of a single image. It is most useful for combining the groups from a GHRS or FOS rapid-readout image into a single spectrum with the full integration time. In addition to summing or averaging the groups of a group format image, **rcombine** will also (if requested) produce a data quality file and a statistical error file for the output image. The output data quality file is produced by propagating the maximum value of the input data quality values from the input groups, and the statistical error file is produced by propagating the statistical errors from the input groups in quadrature. Figure 2.20 shows an example of using the **rcombine** task. In this example, all of the groups from the original FOS rapid mode spectrum are averaged together to produce the final spectrum, and an output data quality and error file is produced.

rcombine does not produce a wavelength file; you can use the wavelength information from the original `.c0h` file for FOS or GHRS spectra.

Figure 2.20: Combining all Groups in FOS Image Using rcombine



Addition With Wavelength Alignment

The tasks **poffsets** and **specalign** can be used together to align and add spectra that are shifted in wavelength or in pixel space with respect to one another. The

poffsets task determines the shift between the input spectra (which can either be the groups of data in a single image or several different images) and writes the shifts to an output table. The shifts can be determined either (1) using the wavelength information in the `.c0h` input files or (2) by cross-correlating features in the input spectra. The **specalign** task combines the spectra using the shift information provided by **poffsets**. For GHRS FP-SPLIT data, in which each group is a convolution of the spectrum with the photocathode response function, **specalign** can also be used to derive the photocathode response function (which is the constant background on which the spectral features are shifted) when it coadds the spectra. These two tasks are most commonly used to combine the groups in an FP-SPLIT GHRS observation, however, they are also useful when combining any spectra that may be shifted in wavelength or pixel space relative to one another.

Using poffsets

The **poffsets** task determines the shift between the input spectra and writes the shifts to an output table. In its default mode, the task determines the shifts by cross-correlating features in the input spectra. However, if the signal-to-noise ratio of the input spectra is too low, **poffsets** will be unable to produce a good solution. In that case you should align the spectra using the wavelength information in the `.c0h` files. This limits the accuracy of the alignment to the accuracy of the zero point of the wavelength calibration for the input images To use **poffsets** to determine the shifts by reading the wavelength information provided in the `.c0h` file (i.e., and not by correlating features in the individual input spectra), set the parameter `usecorr=no` in **poffsets** before running the task.

Using specalign

The **specalign** task uses the output shift table produced by **poffsets** to shift and co-add the input spectrum. In its default mode **specalign** produces:

- An output file containing the combined flux spectrum.
- An output file containing the wavelength solution for the combined spectrum.
- A file containing the constant background relative to which the input spectrum were shifted (i.e., for GHRS FP-SPLIT mode data, the photocathode response function).

If you wish to insert the wavelength calibration information into the header of the combined flux spectrum using World Coordinate System (WCS) keywords, rather than producing a separate file containing the wavelength data, set the parameter `wavelength` to “WCS”.

If you do not wish to produce a constant background file (if for example you are working with low signal-to-noise GHRS FP-SPLIT spectra or in cases where it is inappropriate), set the parameter `niter=0` in **specalign**.

Example of poffsets and specalign

Below we provide an example of using **poffsets** and **specalign** (Figure 2.21, with input and output spectra shown in Figure 2.22). In this example, we combine the groups in an FP-SPLIT GHRS observation into a single spectrum by cross correlating features in the input spectra. The inputs to **poffsets** are the input spectrum and the name of the output table into which the shift information is written. We then use **specalign** in its default mode to produce a combined spectrum and a photocathode response function. The inputs to **specalign** are the name of the shift table produced by **poffsets**[12], the name of the output flux spectrum, the name of the output wavelength spectrum and the name of the output granularity response function. Additional information is available in the online help for the **poffsets** and **specalign** tasks.

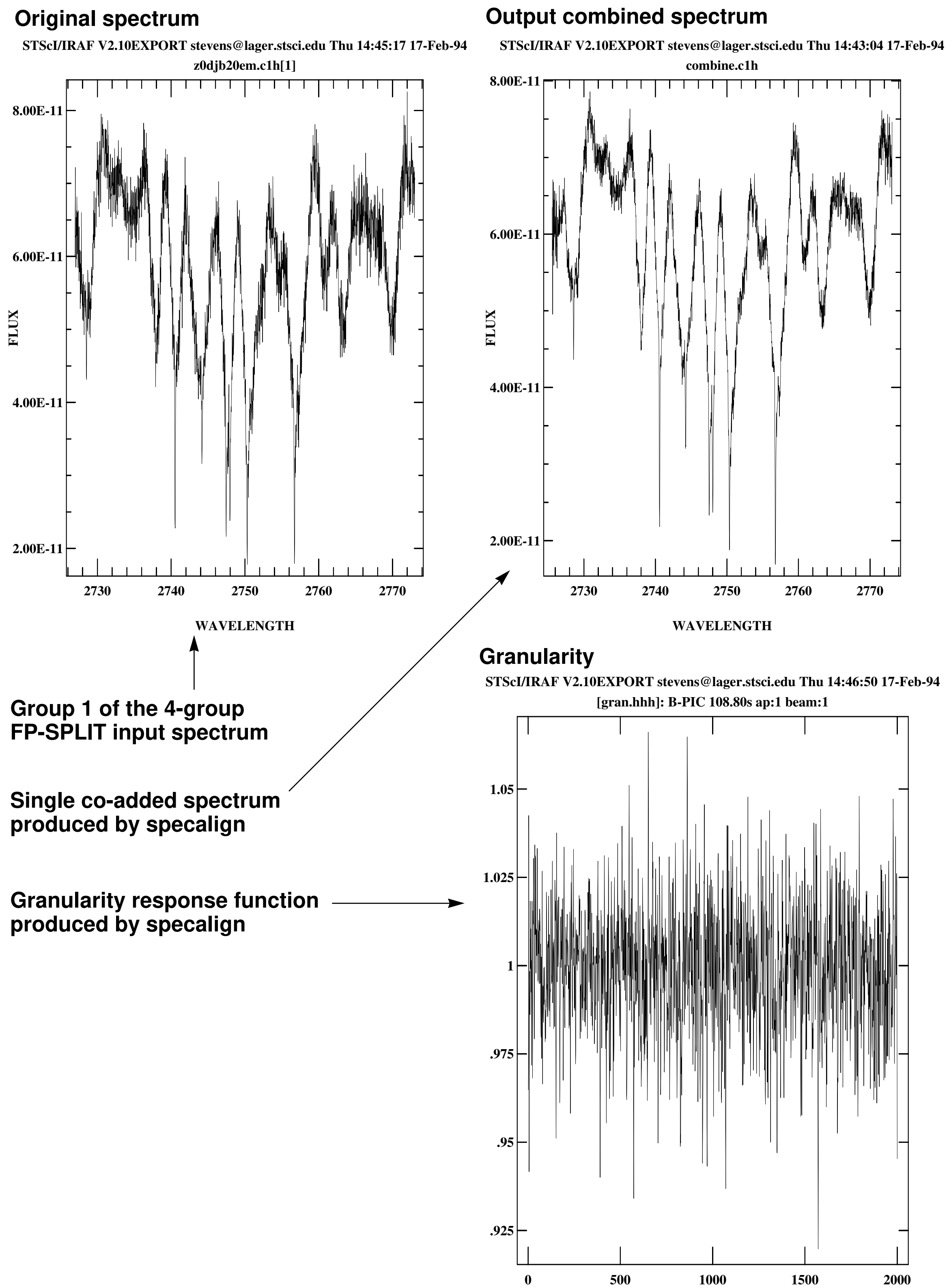
The granularity ideally represents the response of the GHRS detector photocathode. Because it is a response function, the granularity should always be between zero and one. With low S/N data however, the granularity solution may have values greater than 1.5 or 2. This indicates that there was not enough signal to compute a proper granularity.

Figure 2.21: Using poffsets and specalign

```
ct> poffsets z0djb20em.c1h shift.tabs
Searching for shift of data z0djb20em.c1h[1]:
    predicted shift is 0
    section 1 has shift -1.521558236818876E-4.
    Final average = -1.521558236818876E-4, sigma = INDEF.
Searching for shift of data z0djb20em.c1h[2]:
    predicted shift is 19
    section 1 has shift 21.19624414244704.
    Final average = 21.19624414244704, sigma = INDEF.
Searching for shift of data z0djb20em.c1h[3]:
    predicted shift is 39
    section 1 has shift 37.58474906473953.
    Final average = 37.58474906473953, sigma = INDEF.
Searching for shift of data z0djb20em.c1h[4]:
    predicted shift is 59
    section 1 has shift 61.68598931149682.
    Final average = 61.68598931149682, sigma = INDEF.
ct> specalign shift.tabs combine.c1h combine.c0h gran.hhh
ct> 
```

12. Which itself contains the name of the input spectra to be coadded.

Figure 2.22: Input and Output Spectra and Granularity



Combining Wavelength and Flux Information

Calibrated spectra are composed of separate wavelength (`.c0h`) and flux (`.c1h`) files. Before you can use most IRAF and STSDAS tasks (or tasks in other reduction packages) to analyze your data, you need to put wavelengths or dispersion information into the header of the flux file. There are several different ways to do this. The most appropriate method will depend on what type of analysis you wish to do with your data and which tasks you'll be using because not all tasks accept the same types of wavelength information. Some of the methods described here are:

- Using **mkmultispec**: This is the best choice for most users because it preserves all flux information and is supported by many IRAF tasks and other packages.
- Using STSDAS tables: This is the best choice for users who wish to work with analysis tools such as the STSDAS **fitting** package.
- Using the **resample** task: This method loses information, but is simple and provides a format that is easily exported to other software.

Mkmultispec

The most elegant method of combining wavelength and flux information, and one that has no affect on the flux data at all, is to use the **mkmultispec** task to put IRAF multispec-format World Coordinate System (WCS) information into the headers of your flux files (a detailed discussion of the simple linear and more complex multispec coordinate systems is available by typing `help onedspec.package` while in IRAF). The multispec coordinate system is intended to be used with spectra having nonlinear dispersions or images containing multiple spectra; the format is recognized by all tasks in IRAF V2.10 or later.

mkmultispec can put wavelength information into the flux header files in two different ways. The first involves reading the wavelength data from the `.c0h` file, fitting the wavelength array with either Legendre, Chebyshev, or spline functions, and then storing the derived function coefficients in the flux header file (`.c1h`) in the multispec format. A choice of either a Legendre, Chebyshev, or cubic spline (spline3) fitting function of order 4 or larger will produce essentially identical results, all having rms residuals less than 1.0E-4 Å, which is much smaller than the uncertainty of the original wavelength information. Because the fits are so accurate, it is usually not necessary to run the task in interactive mode to examine them.

Because **mkmultispec** can only fit simple types of polynomial functions to wavelength data, this method will *not* work well with FOS prism data, because of the different functional form of the prism mode dispersion solution. For prism spectra, use the header table mode of **mkmultispec** (see below) or the STSDAS tables method.

The other method by which **mkmultispec** can incorporate wavelength information into a flux file is to simply read the wavelength data from the `.c0h` file and place the entire data array directly into the header of the flux (`.c1h`) file. This method simply dumps the wavelength value associated with each pixel in the spectrum into the flux header and is selected by setting the parameter `function=table`. To minimize header size, set the parameter `format` to a suitable value, for example, using `format=8.7g` will retain the original 7 digits of precision of the wavelength values, while not consuming too much space in the flux header file.

Tables

Another way to use wavelength information with your flux spectra is to create an STSDAS table from your spectra; the table will contain columns of wavelength, flux, and (optionally) error values. This can be done using the **imtab** (image to table) task in the STSDAS **ttools** package. This method is necessary if you plan to use certain tasks—such as those in the STSDAS **fitting** package—which do not (yet) recognize the multispec format WCS header information. Figure 2.23 shows how you can easily create a table that contains wavelength, flux, and error data from a calibrated FOS dataset.

Figure 2.23: Combining Flux and Wavelength Using Tables

```
cl> imtab y0cy0108t.c0h[8] y0cy0108t.tab wavelength
cl> imtab y0cy0108t.c1h[8] y0cy0108t.tab flux
cl> imtab y0cy0108t.c2h[8] y0cy0108t.tab error
```

This example is for an ACCUM mode observation where the last data group is number 8 and will produce a table containing three columns, labeled “wavelength”, “flux”, and “error”.

This method of storing spectral information in tables is also the best option if you want to join two or more spectra that were taken with different gratings into a single spectrum that covers the complete wavelength range of all of the individual spectra. Because the data are stored as individual wavelength-flux pairs, there is no need to first resample (and therefore degrade) the individual spectra to a common (linear) dispersion scale before joining them. For example, you could first create separate tables for the spectra from different gratings, and then append the two tables using the **tmerge** task:

```
cl> tmerge n5548_h13.tab,n5548_h19.tab \
>>>  n5548.tab append
```

Note that you will first have to edit out any regions of overlapping wavelength from one or the other of the input tables so that the output table will be monotonically increasing (or decreasing) in wavelength.

Using resample

The method that produces the simplest—and therefore most widely used—wavelength information is to use the **resample** task which will resample your flux data onto a user-selected linear wavelength scale and insert values for the image header keywords CRVAL1 (starting wavelength) and CD1_1 (wavelength increment per pixel). Use of these particular keywords provides some degree of portability between systems because these are standard FITS keywords.

This method has the most severe side-effects in that you not only lose information in your flux spectrum by resampling, but you also lose all further use of the accompanying error (`.c2h`) and data quality (`.cqh`) files which are not, and can not, be resampled.

Spectral Analysis and Manipulation

There are may IRAF and STSDAS tasks available that can be used to plot, manipulate, and fit features in your spectra. Tables 2.11 through 2.13 lists the most relevant tasks, the package in which they are found, the type of input they expect (e.g., flux spectra with wavelength information in WCS), and whether they will work with more than one group in a multiple group image and use the statistical error files in their analysis.

In the next section we describe three particular tasks or packages that you may wish to use to analyze your data:

- **splot** in the **noao.onedspec** package.
- STSDAS **fitting** package.
- **specfit** in the STSDAS **contrib** package.

Table 2.11: Tasks for Fitting Spectra

Task	Package	Purpose
fitprofs	noao.onedspec	Non-interactive Gaussian profile fitting to features in spectra and image lines
nfit1d	stsdas.analysis.fitting	Fit 1-D nonlinear functions to spectra; uses wavelength & error for list/tbl.
ngaussfit	stsdas.analysis.fitting	Fit multiple 1-D Gaussians to spectra; uses wavelength & error for list/tbl.
sfit	noao.onedspec	Fit spectra with polynomial function (WCS header wavelength)
specfit	stsdas.contrib	Fit multiple line profiles and continua to spectra
splot	noao.onedspec	Fit multiple 1-D Gaussians and continua to spectra

Table 2.12: Tasks for Plotting Spectra

Task	Package	Purpose
bplot	noao.ondspec	Plot spectra non-interactively; reads WCS header for wave scale
fwplot	stsdas.hst_calib.ctools	Plot flux vs. wavelength for a single group of an FOS or GHRS image. Will plot errors from statistical error file.
grplot	stsdas.graphics.stplot	Plot arbitrary lines from 1-D image; overplots multiple groups; no error or wavelength information is used
grspec	stsdas.graphics.stplot	Plot arbitrary lines from 1-D image; stacks groups and reads WCS header for wavescale
modeone	stsdas.hst_calib.fos	Restore and display FOS “Mode I” image
rapidlook	stsdas.hst_calib.ctools	Create and display a 2-D image of stacked 1-D images
sgraph	stsdas.graphics.stplot	Plot spectra and image lines; allows overplotting of error bars and access to wavelength array if list or table input is used
specplot	noao.onedspec	Stack and plot multiple spectra; reads wavelength in WCS header

Table 2.13: Utility Tasks for Working with Spectra

Task	Package	Purpose
boxcar	images	Boxcar smooth a list of images; processes one group at a time
continuum	noao.onedspec	Continuum normalize spectra; processes one group at a time
gcopy	stsdas.toolbox.imgtools	Copy multigroup images
gimpcor	stsdas.hst_calib.fos	Calculate and display FOS GIMP values
grlist	stsdas.graphics.stplot	List file names including groups for all groups in image; used to make lists for tasks that do not use group syntax
magnify	images	Interpolate spectrum on finer (or coarser) pixel scale
poffsets	stsdas.hst_calib.ctools	Determine pixel offsets between shifted spectra
rcombine	stsdas.hst_calib.ctools	Combine (sum or average) groups in a 1-D image with option of propagating errors and data quality values
resample	stsdas.hst_calib.ctools	Resample FOS and GHRS data to a linear wavelength scale
sarith	noao.onedspec	Spectrum arithmetic
scombine	noao.onedspec	Combine spectra; reads WCS header keywords for wavelength
specalign	stsdas.hst_calib.ctools	Align and combine shifted spectra (see poffsets)
unwrap	stsdas.hst_calib.fos	Remove *wrap* from FOS data exceeding internal counter limit

splot

The **splot** task in the IRAF **noao.onedspec** package is a good general analysis tool that can be used to examine, smooth, fit, and do simple arithmetic operations on spectra. Figure 2.24 shows a sample plot produced by **splot**. Like all IRAF tasks, **splot** will only work on one group at a time from a multigroup file. You can specify which group is to be operated on by using the square bracket notation, for example:

```
cl> splot y0cy0108t.c1h[8]
```

If you don't specify a group in brackets, the task will take the first group. In order to use **splot** to analyze your HST spectrum, you will first need to write the wavelength information from your `.c0h` file to the header of your `.c1h` files in WCS, using the **mkmultispec** task (see "Mkmultispec" on page 61).

The **splot** task is complex with *many* available options (which are described in detail in the online help within IRAF; type `help splot`). Table 2.14 summarizes some of the more useful capabilities of **splot**. When you are using **splot**, a log file is saved that contains results produced by the equivalent width or de-blending functions. To specify a file name for this log file, set the `save_file` parameter (see the online help).

Figure 2.24: Using splot to Fit Gaussian to an Absorption Line

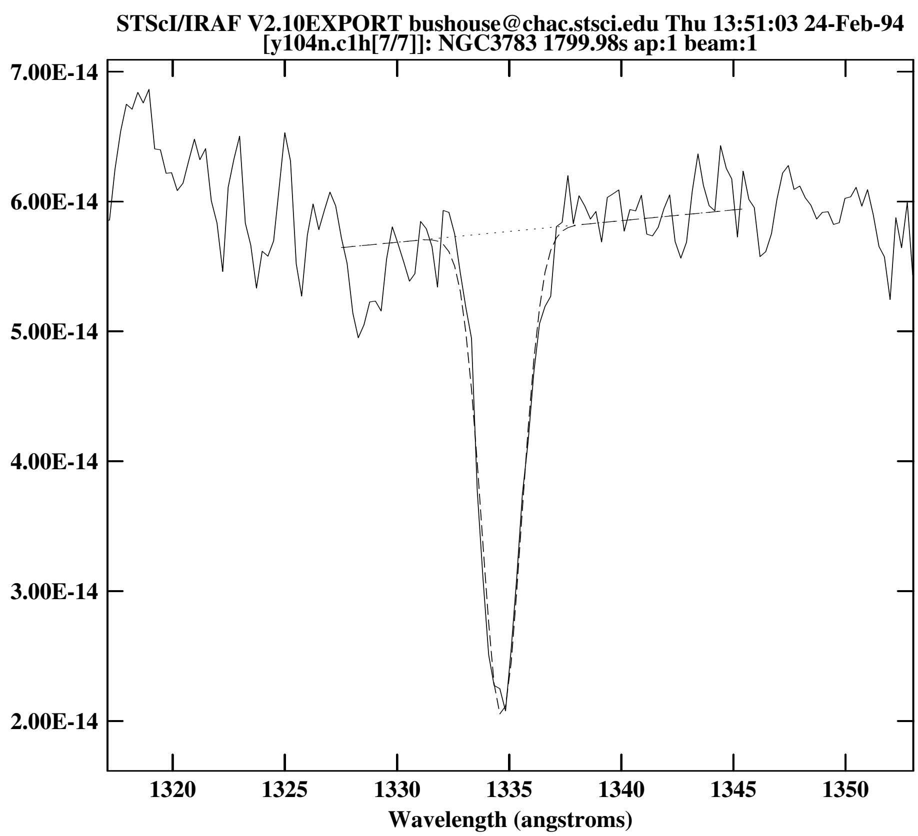


Table 2.14: Capabilities of splot Task

Keystroke Command	Purpose
Manipulating spectra	
f	Arithmetic mode; add and subtract spectra
l	Convert spectrum from F_ν to F_λ
n	Convert spectrum from F_λ to F_ν
s	Smooth with a boxcar
u	Define linear wavelength scale using two cursor markings
Fitting spectra	
d	Mark two continuum points & de-blend multiple Gaussian line profiles
e	Measure equivalent width by marking points around target line
h	Measure equivalent width assuming Gaussian profile
k	Mark two continuum points and fit a single Gaussian line profile
m	Compute the mean, rms, and S/N over marked region
t	Enter interactive curve fit function (usually used for continuum fitting)
Displaying and redrawing spectra	
a	Expand and autoscale data range between cursor positions
b	Set plot base level to zero
c	Clear all windowing and redraw full current spectrum
r	Redraw spectrum with current windowing
w	Window the graph
x	Etch-a-sketch mode; connects two cursor positions
y	Overplot standard star values from calibration file
z	Zoom graph by a factor of two in the x direction
$	Switch between physical pixel coordinates and world coordinates
General file manipulation commands	
?	Display help
g	Get another spectrum
i	Write current spectrum to new or existing image
q	Quit and go on to next input spectrum

STSDAS fitting Package

The STSDAS **fitting** package contains several powerful and flexible tasks for fitting and analyzing spectra. The **ngaussfit** and **nfit1d** tasks, in particular, are very good for interactively fitting multiple Gaussians and nonlinear functions, respectively, to spectral data. Unfortunately, these tasks do not yet recognize the "multispec" WCS method of storing dispersion information. They do recognize the simple sets of dispersion keywords such as W0, WPC and CRPIX, CRVAL, and CDELT, but these forms only apply to linear coordinate systems and therefore would require your data to first be resampled onto a linear wavelength scale before being used. Fortunately, these tasks will accept input from STSDAS tables, in which you can store the wavelength and flux data value pairs or wavelength, flux, error value triples.

When using the **ngaussfit** and **nfit1d** tasks, it is necessary to specify initial guesses for the function coefficients before a fit can be computed. The initial guesses can be specified either via parameter settings in the task's psets or interactively once the task has started up. For example, let's say you want to fit several features using the **ngaussfit** task. Using the default parameter settings you can start the task using the following command line:

```
fi> ngaussfit n4449.hhh linefits.tab
```

This reads spectral data from the image `n4449.hhh` and stores the results of our line fits in the STSDAS table `linefits.tab`.

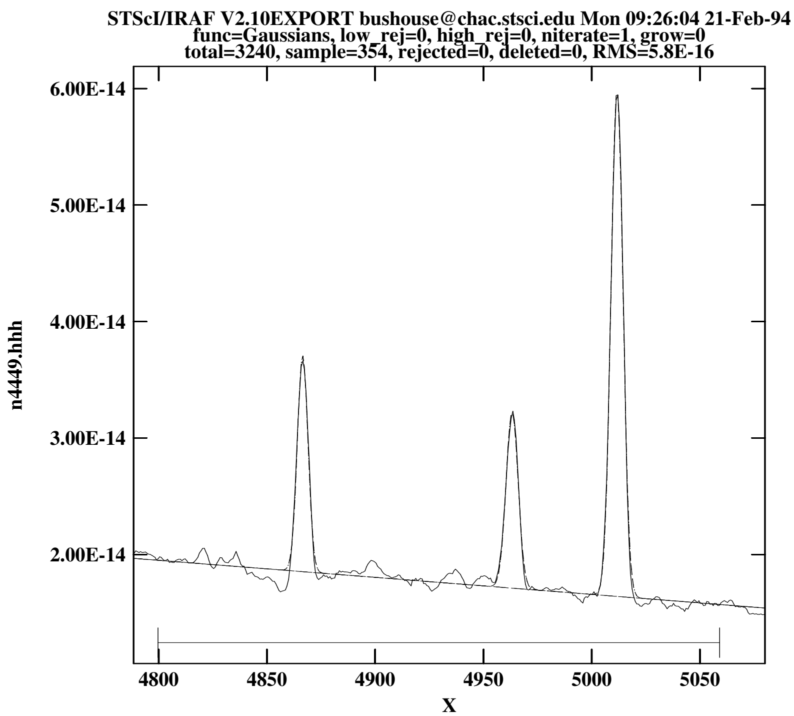
At this point a plot of your spectrum should appear in a plot window and the task will be left in cursor input mode. You can use the standard IRAF cursor mode commands to re-window the plot to restrict the field of view to the region around a particular feature or features that you want to fit. It's also good at this point to:

- Define a sample region (using the cursor mode S command) over which the fit will be computed so that fitting time will not be wasted trying to fit the entire spectrum.
- Define an initial guess for the baseline coefficients by placing the cursor at two baseline locations (one on either side of the feature to be fitted) using the B keystroke.
- Use the R keystroke to redraw the screen and see the baseline that you've just defined.
- Set the initial guesses for the Gaussian centers and heights by placing the cursor at the peak of each feature and typing P.
- Once you've marked all the features that you want to fit, press F to compute the fit.
- The results will automatically be displayed. You can use the `:show` command to see the coefficient values.

Note that when the **ngaussfit** task is used in this way (i.e., starting with all default values), the initial guess for the FWHM of the features will be set to a value of 1. Furthermore, this coefficient (as well as the coefficients defining the baseline) are by default held fixed during the computation of the fit, unless you explicitly tell the task through cursor *colon* commands[13] to allow these coefficients to vary. Actually, it is sometimes best to leave these coefficients fixed during an initial fit, and then allow them to vary during a second iteration. This rule of thumb also applies to the setting of the `errors` parameter which controls whether or not the task will estimate error values for the derived coefficients. Because the process of error estimation is very CPU-intensive, it is most efficient to leave the error estimation turned off until you've got a good fit, and then turn the error estimation on for one last iteration.

Figure 2.25 shows the results of fitting the Hβ (4861Å) and [OIII] (4959 and 5007 Å) emission features in the spectrum of NGC 4449. The resulting coefficients and error estimates (in parentheses) are shown in Figure 2.26.

Figure 2.25: Fitting Hβ and [OIII] Emission Features in NGC 4449



13. See the online help for details and a complete listing of cursor mode colon commands: type `help cursor`.

Figure 2.26: Coefficients and Error Estimates

```
function = Gaussians
coeff1 = 8.838438E-14    (0.)             - Baseline zeropoint (fix)
coeff2 = -1.435682E-17   (0.)              - Baseline slope (fix)
coeff3 = 1.854658E-14   (2.513048E-16)  - Feature 1: amplitude (var)
coeff4 = 4866.511        (0.03789007)     - Feature 1: center (var)
coeff5 = 5.725897        (0.0905327)      - Feature 1: FWHM (var)
coeff6 = 1.516265E-14   (2.740680E-16)  - Feature 2: amplitude (var)
coeff7 = 4963.262        (0.06048062)     - Feature 2: center (var)
coeff8 = 6.448922        (0.116878)       - Feature 2: FWHM (var)
coeff9 = 4.350271E-14   (2.903318E-16)  - Feature 3: amplitude (var)
coeff10 = 5011.731       (0.01856957)     - Feature 3: center (var)
coeff11 = 6.415922       (0.03769293)     - Feature 3: FWHM (var)
rms           = 5.837914E-16
grow          = 0.
naverage      = 1
low_reject    = 0.
high_reject   = 0.
niterate      = 1
sample        =  4800.132:5061.308
```

specfit

The **specfit** task, in the STSDAS **contrib** package, is another powerful interactive facility for fitting a wide variety of emission and absorption line, and continuum models to a spectrum. This task was written by Gerard Kriss at Johns Hopkins University and because it is a contributed task little to no support is provided by the STSDAS group. There is extensive online help available, however, which should usually be sufficient to guide you through the task.[14]

The input spectrum to **specfit** can be either an IRAF image file or an ASCII file with a simple three-column (wavelength, flux, and error) format. If the input file is an IRAF image, the wavelength scale is set using values of W0 and WPC or CRVAL1 and CDELT1. Hence, for image input, the spectral data must be on a linear wavelength scale. In order to retain data on a non-linear wavelength scale, it is necessary to use the ASCII file mode of input, where the wavelength values associated with each data value can be specified explicitly. The help file explains a few pieces of additional information that must be included as header lines in an input text file.

By selecting a combination of functional forms for various components, you can fit complex spectra with multiple continuum components, blended emission and absorption lines, absorption edges, and extinction. Available functional forms include linear, powerlaw, broken powerlaw, blackbody, and optically thin recombination continua, various forms of Gaussian emission and absorption line and absorption edge models, Lorentzian line profiles, damped absorption line profiles, and mean galactic extinction.

14. Additional information is available in the *Astronomical Data Analysis Software and Systems III*, ASP Conference Series, Vol. 61, page 437, 1994.

HSP Specific Tasks

There are a few tasks in STSDAS that may be useful for conditioning and analyzing data from the High Speed Photometer. Table 2.15 summarizes these tasks.

Table 2.15: HSP Tasks in STSDAS

Task	Function
parthitv	Remove particle event.
polyepoch	Fit polynomials to coefficients as functions of the epoch.
polyfit	Fit a polynomial to a specified quantity
posvel	Calculate position and velocity vector of the spacecraft
twodpolyfit	Two-dimensional polynomial fit of temperature and epoch

Getting IRAF and STSDAS

Both IRAF and STSDAS are provided free of charge to the astronomical community. You must have IRAF to run STSDAS. Detailed information about installing and retrieving STSDAS is found in the *STSDAS Site Manager's Installation Guide and Reference*. If you have any problems getting and installing STSDAS, TABLES, or any other packages or data described in this handbook, please contact the Help Desk by sending e-mail to: `help@stsci.edu`.

A description of how to install the **synphot** data files is provided on page 71.

Retrieving the IRAF and STSDAS Software

If you already have IRAF and STSDAS installed on your system, you should skip this section and look at "Setting Up IRAF" on page 14.

There are three ways to get the software:

- Use the World Wide Web.
- Use anonymous FTP.
- Request a tape.

World Wide Web

When you access the STSDAS World Wide Web page, you will see an option for getting the software. Links and instructions are provided to download the appropriate files to your local system or to display the software directory, from which you can select the series of smaller files. The STSDAS web page is available at the following URL: `http://ra.stsci.edu/STSDAS.html`

Anonymous FTP

- **IRAF**: `iraf.noao.edu` (140.252.1.1)
- **STSDAS**: `ftp.stsci.edu` (130.167.1.2)

There are two points to remember when using FTP to retrieve STSDAS:

- You must retrieve and install the TABLES package before STSDAS.
- You should retrieve the `README` file from the directory `/software/ stsdas/v1.3` and read it to find out which files you should retrieve.

You must have IRAF installed on your system to install TABLES and STSDAS. When you retrieve STSDAS, you must also retrieve the TABLES package, and TABLES must be installed first.

Instructions for installing STSDAS are available in the `doc` subdirectory of the directory where you find STSDAS. The complete instructions for installing STSDAS, TABLES, and all of the supporting software and reference files (including instrument reference files and the **synphot** dataset) are found in the *STSDAS Site Manager's Installation Guide and Reference*.

Requesting Tapes

You can ask to have the software shipped to you on magnetic tape in a variety of formats. To do so, you will need to contact the Help Desk by sending e-mail to `help@stsci.edu`. The Help Desk staff will send you an ASCII text version of the STSDAS Software Request Form, which you can then complete and return via e-mail. The software can also be registered and requested using on-line forms available through World Wide Web at the following URL:

`http://ra.stsci.edu/RegistForm.html`

When you request the STSDAS software, you can also ask for the appropriate version of IRAF, which will be requested for you— simply check the appropriate box on the form under "Do You Already Have IRAF Installed?" If you prefer to request the IRAF software independent of STSDAS, you can do so by sending e-mail to: `iraf@iraf.noao.edu`

Synphot Dataset

This manual sometimes refers to the **synphot** dataset, which must be available in order to run tasks in the STSDAS **synphot** package. These data files are not included with the STSDAS software and must be retrieved independently. To do this, you need to retrieve a series of compressed tar files from the STEIS directory `software/stsdas/refdata/synphot`. After uncompressing and extracting the tar files (as was described in the previous sections), you need to unpack the FITS files as described below.

The synthetic photometry data are read in similar way as the instrument datasets, using the script `unpack.cl` provided in the top directory. This script is run within

IRAF to convert data from FITS format into the format used by the **synphot** task. This script assumes you have the logical `crrefer` set up in your `extern.pkg` file (which is in the directory `$iraf/unix/hlib` (Unix) or `$iraf/vms/hlib` (VMS)) or have it set up in your session. You do this by placing the command below in `extern.pkg` or by typing it on the command line:

```
set crrefer = "/node/partition/stdata/synphot/"
```

Figure 2.27 shows how to convert the files.

Figure 2.27: Unpacking Synthetic Photometry Files

```
% cl
cl> cd /node/partition/stdata/synphot
cl> set crrefer = "/node/partition/stsdata/synphot/"
cl> task $unpack = unpack.cl
cl> tables
ta> fitsio
fi> unpack
```

Just in case...

The "$" is used because the task has no parameter file

Note that all three **synphot** files must be unloaded for the script to complete successfully.

References

Available from STScI

- *STSDAS Users Guide*, version 1.3.3, September 1994.
- *STSDAS Installation and Site Managers Guide*, version 1.3.3, March 1995
- *Synphot Users Guide*, version 1.3.3, March 1995.
- *IGI Reference Manual*, version 1.3, October 1992.

Available from NOAO

- *A Beginners Guide to Using IRAF,* 1994, J. Barnes.
- *User Manual for SAOimage,* 1991, M. Van Hilst.
- *Photometry Using IRAF*, 1994, L. Wells.
- *A User's Guide to Stellar CCD Photometry with IRAF*, 1992, P. Massey and L. Davis.

Other References Cited in This Chapter

- Horne, K., 1988, in *New Directions in Spectrophotometry*, A.G.D. Philip, D.S. Hayes, and S.J. Adelman, eds., L. Davis Press, Schenectady NY, p. 145.
- Koorneef, J., R. Bohlin, R. Buser, K. Horne, and D. Turnshek, 1986, in *Highlights of Astronomy*, Vol. 7, J.-P. Swinds, ed., Reidel, Dordrecht, p. 833.
- Kriss, G., 1994, in *Astronomical Data Analysis Software and Systems III*, PASP Conference Series, Vol. 61, p. 437.

CHAPTER 3:
Data Tapes and File Structures

In This Chapter...

Within a few weeks of your HST observations STScI will mail to you one or more tapes containing your data. If you have just received your shipment of HST observation data, you will find that your package includes:

- At least one data tape, together with a tape log listing its contents.
- Several printouts.

This chapter explains how to use the tape log to understand what kind of files are on the tape, how to understand what the hardcopy printouts represent, and how to actually read a tape. A brief outline of the file structure is given as well.

If you are an archival user, you should also refer to Chapter 4 for information about data retrieval and Archive structure.

The printouts that you receive may be substantially different from those described here. Institute staff are redesigning the paper products to make them more complete and self-explanatory. This document describes the documents as they were in December 1995.

Tape Log and Contents

With your tape you receive a *tape log* which lists the contents of your tape. A sample tape log is reproduced in Figure 3.1. There are at least three types of files on any tape; these files will be described in more detail in the following sections:

- An ASCII *trailer file* (page 78), written as a FITS-format ASCII table.
- A series of *FITS-format data files* which comprise your HST data (page 78).
- A *post-observation data quality* (PDQ) file (page 79), written as a FITS-format ASCII table.
- An OMS *jitter image* file and *jitter table*.

Tapes containing data from Faint Object Spectrograph (FOS) and Goddard High Resolution Spectrograph (GHRS) observations will include additional FITS-format tables and images containing the reference files used by the Routine Science Data Processing (RSDP) pipeline to calibrate the data.

Tape Log

The tape log should list, for each observation, a series of data files including the *trailer* file (identified by the extension `.trl`). All files that have the same rootname (e.g., `u2ri3101t`) belong to a single observation. Jitter image files have the same first 8 characters, but the ninth character is a “j”.[1]

The first letter of the rootname of the dataset indicates the instrument with which the data were observed:

- FOC data begin with “x”.
- FOS data begin with “y”.
- FGS data have rootnames beginning with “f”.
- GHRS data begin with “z”.
- HSP data begin with “v”.
- WF/PC-1 data begin with “w”.
- WFPC2 data begin with “u”.

The example in Figure 3.1 lists the files for two Wide Field Planetary Camera 2 (WFPC2) datasets, each containing twelve files.

1. Although this example shows a WF/PC-1 tape log, it illustrates the relevant features of tape logs provided with any HST dataset.

Figure 3.1: Sample Tape Log

```
Space Telescope Science Institute Data Archive and Distribution Service.

This is the tape log file indicating the content and the structure of the
files on this tape.

This tape is in FITS format and uses standard FITS TABLES extension that
your FITS reader should be capable of handling.

If you don't have this capability, you can get a copy of the TABLES package
that would run under IRAF from the anonymous ftp.stsci.edu service in the
directory 'software/tables'.

This file first provides a list of all the science datasets (observations)
on the tape (if there are any present).  A file by file listing of the files
on the tape in the order of their occurrence is provided.  All FITS files are
written first and any binary non_FITS files are written at the end of the tape.

--------------------------------------------------------------------------------
 This tape contains the following science datasets.  Files from non-science
 datasets, for example calibration reference files, may also be on the tape.
--------------------------------------------------------------------------------
U2RI3101  wfpc2 observation of target SGP014               , exposed for
             26.0000000000  seconds, taken in mode IMAGE             and
          configuration WFPC2            on 28-AUG-1995   06:40:16  , using
          spectral elements F850LP              .

U2RI3102  wfpc2 observation of target SGP014               , exposed for
             26.0000000000  seconds, taken in mode IMAGE             and
          configuration WFPC2            on 28-AUG-1995   06:42:16  , using
          spectral elements F850LP              .

--------------------------------------------------------------------------------
 This tape contains the following FITS files
--------------------------------------------------------------------------------
FILE   NAME AND DESCRIPTION                  GENERATED    SIZE(KBYTES)  DATATYPE
1      AUTO4358_1.LOG                   15-SEP-1995 12:56:47. 29     unsigned*2
       Tape Log File

2      U2RI3101T.C0F                    28-AUG-1995 19:27:35  10043  real*4
       Calibrated Science Image File    TARGET: SGP014

3      U2RI3101T.C1F                    28-AUG-1995 19:27:35  5040   integer*2
       Data Quality File                TARGET: SGP014

4      U2RI3101T.C3T                    28-AUG-1995 19:27:35  87     unsigned*2
       Throughput table derived from SYN TARGET: SGP014

5      U2RI3101T.D0F                    28-AUG-1995 19:27:35  5040   integer*2
       Science Image File               TARGET: SGP014

6      U2RI3101T.Q0F                    28-AUG-1995 19:27:35  5040   integer*2
       Data Quality File, Scienc Image F TARGET: SGP014

7      U2RI3101T.Q1F                    28-AUG-1995 19:27:35  101    integer*2
       Data Quality File, Extracted Engi TARGET: SGP014

8      U2RI3101T.SHF                    28-AUG-1995 19:27:35  33     unsigned*2
       Standard Header Packet           TARGET: SGP014

9      U2RI3101T.TRL                    28-AUG-1995 19:27:35  16     unsigned*2
       Trailer File                     TARGET: SGP014

10     U2RI3101T.X0F                    28-AUG-1995 19:27:35  101    integer*2
       Extracted Engineering File       TARGET: SGP014

11     U2RI3101T.PDQ                    30-AUG-1995 02:23:37  16     unsigned*2
       PODPS Data Quality Comments

12     U2RI3101J.JIF                    29-AUG-1995 20:49:34  25     integer*2
       OMS Jitter Image

13     U2RI3101J.JIT                    29-AUG-1995 20:49:34  25     binary
       OMS Jitter Table

14     U2RI3102T.C0F                    28-AUG-1995 19:32:12  10043  real*4
       Calibrated Science Image File    TARGET: SGP014

15     U2RI3102T.C1F                    28-AUG-1995 19:32:12  5040   integer*2
       Data Quality File                TARGET: SGP014

16     U2RI3102T.C3T                    28-AUG-1995 19:32:12  87     unsigned*2
       Throughput table derived from SYN TARGET: SGP014

17     U2RI3102T.D0F                    28-AUG-1995 19:32:12  5040   integer*2
       Science Image File               TARGET: SGP014

18     U2RI3102T.Q0F                    28-AUG-1995 19:32:12  5040   integer*2
       Data Quality File, Scienc Image F TARGET: SGP014

19     U2RI3102T.Q1F                    28-AUG-1995 19:32:12  101    integer*2
       Data Quality File, Extracted Engi TARGET: SGP014

20     U2RI3102T.SHF                    28-AUG-1995 19:32:12  33     unsigned*2
       Standard Header Packet           TARGET: SGP014

21     U2RI3102T.TRL                    28-AUG-1995 19:32:12  16     unsigned*2
       Trailer File                     TARGET: SGP014

22     U2RI3102T.X0F                    28-AUG-1995 19:32:12  101    integer*2
       Extracted Engineering File       TARGET: SGP014

23     U2RI3102T.PDQ                    30-AUG-1995 02:23:50  16     unsigned*2
       PODPS Data Quality Comments

24     U2RI3102J.JIF                    29-AUG-1995 20:49:38  25     integer*2
       OMS Jitter Image

25     U2RI3102J.JIT                    29-AUG-1995 20:49:38  25     binary
       OMS Jitter Table
```

Trailer Files

The trailer files are ASCII files that contain the log of the processing of your data by the Post-Observation Data Processing System (PODPS) pipeline. The trailer files can be printed after reading from tape.

Note that the trailer file is formatted with 132 columns.

FITS-Format Data Files

Each image is written to tape as a single FITS file which appears as a single entry in the tape log. This file has the nine character rootname uniquely identifying the observation, followed by a three character extension. For FITS images, the last letter in the extension is "f" which distinguishes the FITS-format file from GEIS (Generic Edited Information Set) files, which contain a separate header and data file for each image. The extension is assigned according to the type of data file. This image is a normal FITS format file that is readable by any FITS reader.[2]

- A file with a nine-character rootname followed by "_cvt." for "converted", to distinguish it from GEIS (Generic Edited Information Set) format, followed by a three-character extension. The extension is assigned according to the type of data file. This image is a normal FITS format file that is readable by any FITS reader. In the HST default FITS format, HST group data are stored in an extra dimension of the FITS file.[3] HST data are written to tape in this format in order to keep data constituting a single observation together in one file (e.g., four Wide Field (WF) or Planetary Camera (PC) charge-coupled device (CCD) frames; FOS spectra from ACCUM mode, etc.). The GCOU field on the tape log gives the number of groups associated with an image. (Group format is explained in more detail on page 86).

The DIMENS and BITP (bits per pixel) fields can be used to determine the size of each file. For reference, Table 3.1 below gives the typical sizes (and range of possible sizes) of single files and complete datasets for each instrument.

2. A description of the FITS format and the various options and parameters that can be used in the FITS standard can be found in the document "Implementation of the Flexible Image Transport System (FITS)," by the NASA/OSSA Office of Standards and Technology. The document is available via FTP to nssdca.gsfc.nasa.gov in the directory FITS. A listing of FITS standards and documentation is available via the World Wide Web from NRAO.

3. The HST default FITS format is to use IEEE format with all groups written to a single FITS file and the group parameters written to an ASCII table. Type "help fitsio opt=sys" from within IRAF for details.

Table 3.1: Average Instrument File and Dataset Sizes

Instrument	Mode	Single File Size	Dataset Size
FOC	Normal 512 x 512	1 MB	~4 MB
	Zoom 1024 x 512	2 MB	8-10 MB
FOS	Normal	20-80 KB	< 500 KB
	Rapid readout	> 4 MB	> 10 MB
FGS	POS and Trans	2 MB	6 MB
GHRS	Normal	20-80 KB	< 500 KB
	Rapid readout	> 4 MB	> 10 MB
HSP	Normal	~ 40 KB	~ 500 KB
	Occasional	> 125 MB	> 1 GB
WF/PC-1 and WFPC2	Normal (full mode)	10 MB	26 MB
	Area mode	2.5 MB	7 MB

Printouts of Files

In addition to the trailer files provided on tape, you will receive hardcopies of files that will help you assess the quality of your data. You will also receive a hardcopy plot of a spectrum or photographic image from each observation. Hardcopies may also be provided for:

- PDQ (PODPS Data Quality) files.
- OCX (OSS Observer Comment) files.

These files may also be retrieved from the HST Archive (see Chapter 4).

PDQ Files

The Post Observation Summary and Data Quality Comment files—*PDQ files*—contain predicted as well as actual observation parameters extracted from the standard header and science headers. They also contain a comment on any obvious features in the spectrum or image, as noted by the OPUS data assessor, or information about problems or oddities encountered during the observation or data processing.

A sample PDQ file is reproduced in Figure 3.2. In this example, note the comments at the top, and the reduced exposure time.

Figure 3.2: Sample PDQ File

```
---------- Post Observation Summary and Data Quality Comments (PDQ) ----------

DATASET = 'U2RI3101T'
Proposal Id: 05913                                                  dd/mm/yy
Principal Investigator  : REID        , IAIN NEILL            Date: 28/08/95

Target Name    : SGP014
Target RA (hms):   0 52 52.262         Target Dec (dms): -30  6 35.110

Actual Start Time of Observation : 28/08/95 06:40:16 UT
Calib. Type (Int/Ext/ ):                  Calibration Flag (Y/N) : Y
Operating Mode : IMAGE                    Aperture : PC1

-------------------------- Data Quality Evaluation: --------------------------
QUALITY = 'OK'
QUALCOM1= 'SGP014.  The star is well exposed.'
QUALCOM2= ''
QUALDATE= '28-AUG-1995 19:39'

----------------------- Observation Characteristics: -------------------------

  -- PLANNED characteristics of WFPC2 exposure from SHH file --
TargType: EXTENDED                       Mode    : FULL
Filter 1: F850LP                         Filter 2:
Commanded exposure duration (sec) :     26.00

   -- ACTUAL characteristics of WFPC2 exposure from D0H file --
TargType: EXT                            Mode    : FULL
Filter 1: F850LP                         Filter 2:
Actual exposure duration (sec) :        26.00

---------------------------- D0H Data Structure: -----------------------------
D0H File Group Count: 4             DCF fill: 0         Axis 1 length: 800
                                  PODPS fill: 0         Axis 2 length: 800

-------------------------- Observation Statistics: ---------------------------


                                                    Below 800   Above 4094
     Filename    pixels   mean  stdev   min   max  count   pct  count   pct
U2RI3101T.D0D[1] 639200  315.8    0.0   311  1770 639187  99.9      0   0.0
U2RI3101T.D0D[2] 639200  353.0   30.4   351  4095 639175  99.9      1   0.0
U2RI3101T.D0D[3] 639200  305.1   30.1   303  1830 639194  99.9      0   0.0
U2RI3101T.D0D[4] 639200  310.5   25.0   308  1721 639183  99.9      0   0.0



------------------------------- Extracted OMS Keywords: ----------------------

GUIDECMD= 'FINE LOCK/GYRO      '/ Commanded Guiding mode
GUIDEACT= 'FINE LOCK/GYRO      '/ Actual Guiding mode at end of GS acquisition
NLOSSES =                    0 / Number of loss of lock events
LOCKLOSS=                  0.0 / Total loss of lock time (sec)
NRECENT =                    0 / Number of recentering events
RECENTR =                  0.0 / Total recentering time (sec)
V2_RMS  =                  3.4 / V2 Axis RMS (milli-arcsec)
V2_P2P  =                 20.3 / V2 Axis peak to peak (milli-arcsec)
V3_RMS  =                  6.2 / V3 Axis RMS (milli-arcsec)
V3_P2P  =                 58.1 / V3 Axis peak to peak (milli-arcsec)

-------------------------------  Additional Comments: ------------------------
This observation was taken under single star guidance.  There may be some
motion around a point centered on the guide star position in FGS3.

The distance from the single guide star to the target is 12.1 arc-minutes.
```

OCX Files

The Observer Comment Files—*OCX files*—are produced by the Observation Support System (OSS) personnel. These files are not created for every observation but, when available, they contain updated mission information obtained at the time the observation was executed, and OSS keywords and comments. Prior to April 17, 1992, OCX files were not always archived separately and, in some cases, were prepended to the trailer file (which is on your tape).

After early February 1995, OCX files were produced only when an observation was used to locate the target for an Interactive Target Acquisition. At this time, mission and spacecraft information (like that shown in Figure 3.3) were moved to the PDQ reports and the Observation Logs (OMS jitter image and jitter table). Figure 3.3 is a pre-1995 sample OCX file: note the comments at the end.

Figure 3.3: Sample OCX File

```
******************************************************************************
* *
* This is the data quality evaluation report from OSS operations personnel *
* *
******************************************************************************


 Faint Object Camera (FOC)
 Script for Checking Quality/Utility of Received Science Data

 Observation: Object name: Executed (UT): Received (UT):
 X0KE0101A NGC4486 91.173/ 22:53:44 22-JUN-1991 23:19

 FOC Specific Information
 ------------------------

 Expected Received

 Science Data Mode: image image
 Calibration Mode: _ _
 Camera Used: F/96 f/96
 Data Format: 512 x 1024 512 X 1024
 PCS Mode: FINELOK finelock

 File Information
 ----------------

 D0D D0H SHD SHH ULD ULH
 Expected: Y Y Y Y Y Y
 Received: Y Y Y Y Y Y
 Group Count: 1 1 1

 Data file lost? No Data sent to [DUMP]? No In correct directory? Yes

 Data Information
 ----------------

 Expected exposure time: 1500.000 Sec
 Expected filters: FW1: CLEAR
 FW2: CLEAR
 FW3: CLEAR
 FW4: F372M
 Expected calibration lamp status: NONE

 -------------------------------------------------------------------------
 | Grp | Data | Defects/ | Int. Level | Background | S/N | Focus |
 | # | Dropout | Anomalies * | of Target | Int. Level | Est. | |
 -------------------------------------------------------------------------
 | 1 | n | n | 239 | 15 | good | _ |
 -------------------------------------------------------------------------
 * explain in comments

 Coronagraphic finger used? 0.8" n 0.4" n

 Is the bright object behind the finger? n

 Distant Moving Objects: Field targeted? n

 Any suspicious trails? n


Comments: Target is M87. OSS calculated and sent these offsets:
 (0.157, -1.012). Exposure was disrupted by loss of the
 take data flag. Exposure started five times with a total
 exposure of 603 seconds instead of the scheduled 1500.
 Highest intensity was actually around 275 due to a count
 rollover; (256+19=275).
```

Note Comments

Reading HST Data Tapes

In the previous section we described what you should have received with your shipment of HST data. In this section we expand on that, describing how you can use the tasks in STSDAS to actually read the tape. The data on your tape are in FITS format. In STSDAS, you will read the data onto disk using the **strfits** task. This will create a series of GEIS files. This section describes:

- The general process of reading a tape.
- Preliminary steps, such as mounting the tape and allocating a drive.
- The STSDAS commands to read the tape.

The STSDAS **strfits** FITS reader preserves the multigroup format of an HST image. This format *must* be retained if you plan to recalibrate your data in STSDAS.

To read an HST data tape, you need to:

1. Start IRAF and load the **stsdas** and **fitsio** packages.
2. Mount the tape.
3. Set global parameters.
4. Set the **strfits** parameters and read the tape.

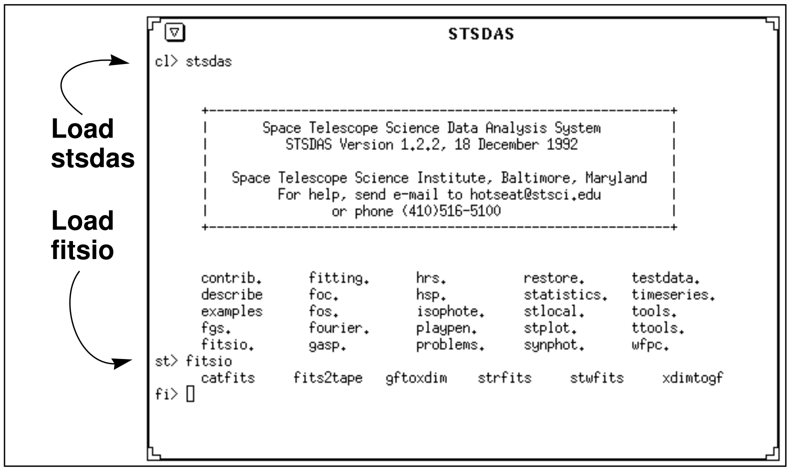
Loading Packages

Go to your IRAF home directory and start IRAF by typing:

```
cl
```

This will start an IRAF session. Software in IRAF is organized into *packages*. To load a package, type its name. Once you are in IRAF, load the **stsdas** and **fitsio** packages as shown in Figure 3.4.

Figure 3.4: Loading Packages in STSDAS



The prompt (such as `fi>`) shows the first two letters of the most recently loaded package. The **fitsio** package contains tasks for handling the FITS format files used for HST images. You can use **catfits** to produce a listing of the contents of your tape, and **strfits** to read the data onto disk. When you are done working with your data, you may choose to write it back out to tape using **stwfits**.

Mounting the Tape

Mount the tape on your tape drive. Allocate the device within IRAF by typing:

`fi> allocate` *device*

where *device* is the IRAF name of the tape drive. If you are not sure how to mount tapes or don't know the IRAF names that match your tape drives, see your local system administrator for help.

Setting Global Parameters

Set `imtype` to specify that the files are to be read in as GEIS format, for example:

```
fi> set imtype="hhh"
```

Go to the directory in which you want the read-in files to be stored. For example:

`fi> cd` */nem/data1/hstdata* ← **Replace this string with your directory path**

Using strfits

Like most IRAF and STSDAS tasks, **strfits** has several parameters that control the task's behavior. To edit the parameters, use the **epar** task[4]:

```
fi> epar strfits
```

In **strfits** you should set:

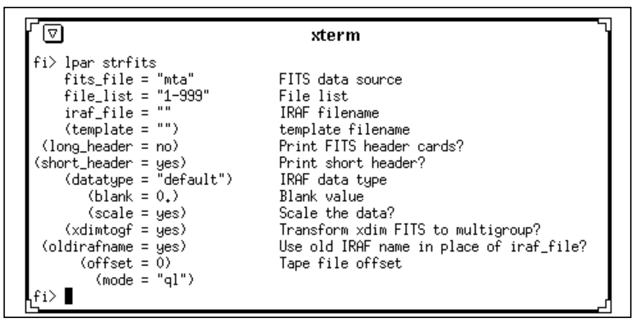
- `fits_file` to the IRAF name of your tape drive.
- `file_list` to specify the files to be read off tape.
- `xdimtogf = yes` to specify that a multigroup GEIS file be created.
- `oldirafname` = yes to restore the original names of files.

When you are finished editing the parameters, type `:go` from within **epar** to automatically run **strfits**. Or type `:q` to return to the IRAF prompt. From the IRAF prompt you can type `strfits` to run the program.

Be sure to set `oldirafname` and `xdimtogf` to “yes” or else the tape will not be read correctly and you will not be able to manipulate the data. This is *vital* if you plan to recalibrate your data.

You can display the current values of the **strfits** parameters by typing `lpar strfits`, as shown in Figure 3.5.

Figure 3.5: Displaying Parameter Values with lpar



```
xterm
fi> lpar strfits
    fits_file = "mta"          FITS data source
    file_list = "1-999"        File list
    iraf_file = ""             IRAF filename
    (template = "")            template filename
 (long_header = no)            Print FITS header cards?
(short_header = yes)           Print short header?
    (datatype = "default")     IRAF data type
       (blank = 0.)            Blank value
       (scale = yes)           Scale the data?
    (xdimtogf = yes)           Transform xdim FITS to multigroup?
 (oldirafname = yes)           Use old IRAF name in place of iraf_file?
      (offset = 0)             Tape file offset
        (mode = "ql")
fi>
```

In this example, the tape drive is “mta” (identified by the `fitsfile` parameter) and files 1 through 999 (or end of tape, whichever occurs first) will be read from the tape and restored using their original file names (because the `oldirafname` parameter is set to “yes”). The parameter `xdimtogf` is set to “yes,” meaning that a multigroup GEIS file will be written.

Once your data are read in, you can deallocate your tape drive by typing:

`fi> deallocate` *device*

4. When the parameter editor comes up, move up or down using the arrow keys and change any values that are necessary. **epar** is described on page 20.

Data Files

Once you have read your tape and gotten a directory listing of your files, you will notice that you have many more files than you have HST exposures (observations).

This section explains how to:

- Identify and understand the nature of each of the data files from an HST exposure.
- Learn about the observational parameters of each exposure by examining the header keywords.

If you used **strfits** to read your data tape as we described on page 83, your data will now be in GEIS format. If you do a directory listing (type `dir` within IRAF), you will find many files, all of which have a nine-character rootname and a three-character extension.

Header and Data Files

Files whose extensions end with the letter "h" (e.g., `w0lo0105t.c1h`) are ASCII *header files*. The header files contain keywords that describe the parameters used to take the observation, the processing of the data, and the properties of the image. Files whose extensions end in the letter "d" (e.g., `w0lo0105t.c1d`) are binary *data files*. A single GEIS image is composed of a header and data *pair* (e.g., the files `w0lo0105t.c1h` and `w0lo0105t.c1d` together represent a single image).

Rootnames and Datasets

By definition, a *dataset* is the collection of all files produced by the Routine Science Data Processing (RSDP) pipeline for a single HST exposure. The files in a dataset all share the same nine-character rootname or *IPPPSSOOT*. As defined in Table 3.2, the rootname follows a specific naming convention that allows each executed observation to be uniquely tied to the scheduling information from which it originated.

Table 3.2: IPPPSSOOT Root File Names

Character	Meaning
`I`	Instrument used, will be one of: ***U*** - Wide Field/Planetary Camera-2 ***V*** - High Speed Photometer ***W*** - Wide Field/Planetary Camera ***X*** - Faint Object Camera ***Y*** - Faint Object Spectrograph ***Z*** - High Resolution Spectrograph ***E*** - Engineering data ***F*** - Fine Guidance Sensors ***H-N*** - Reserved for future instruments ***O*** - Intermediate product files ***S*** - Engineering subset data ***T*** - Guide star position data
`PPP`	Program ID; can be any combination of letters or numbers (46,656 combinations possible). There is a unique association between program ID and proposal ID.
`SS`	Observation set ID; any combination of letters or numbers (1,296 possible combinations).
`OO`	Observation ID; any combination of letters or numbers (1,296 possible combinations).
`T`	Source of transmission (RSDP environment) ***R*** - Real time (not recorded) ***T*** - Tape recorded ***M*** - Merged real time and tape recorded ***N*** - Retransmitted merged real time and tape recorded ***O*** - Retransmitted real time ***P*** - Retransmitted tape recorded

File Extensions

Each file in a dataset has a three-character extension. For each instrument, this extension uniquely identifies the file contents. Examples of file extensions are `.q0h` and `.q0d`. File extensions ending in "h" are image header files in ASCII format and file extensions ending in "d" are actual images in binary format. In this specific example, "q" indicates the data quality file for raw science data. Since the meaning of file extensions varies by instrument, their description is deferred to the individual instrument chapters.

Group Data

The data from a single HST observation are often composed of multiple images or spectra. For example, a single WF/PC-1 exposure is obtained as four images (one image for each CCD chip). Likewise, the FOS and GHRS obtain data in a time-resolved fashion so that a single FOS or GHRS exposure is composed of many spectra—one corresponding to each readout. GEIS files use group format to keep all of the data from a given HST exposure together in a single image file. The data corresponding to each sub-image (for the WF/PC-1 or WFPC2) or each sub-integration (for the FOS or GHRS) are stored sequentially in the groups of a single GEIS image. The header file for an image contains the information that applies to the observation as a whole (i.e., to all the groups in the image). The group-specific keyword information is stored with the group data itself in the binary data file.

The *number* of groups associated with each observation varies with the instrument configuration, observing mode, and observing parameters. In Table 3.3 we list, for the most commonly-used modes of each instrument, the *contents*, and where unambiguous, number of groups in the final calibrated image.

Table 3.3: Groups in Calibrated Images, by Instrument and Mode

Instrument	Mode	Number of Groups	Description
FGS	All	17	FGS data are not reduced with IRAF and STSDAS. Therefore, FGS groups have different meaning than for the other instruments.
FOC	All	1	All FOC images have only a single group.
FOS	ACCUM	*n*	Group *n* contains accumulated counts from groups (subintegrations) 1, 2, ... *n*. The last group is the full exposure.
	RAPID	*n*	Each group is an independent subintegration with exposure time given by group parameter EXPOSURE.
HSP	All	1	HSP datasets always have only a single group that represents either digital star (`.d0h`, `.c0h`), digital sky (`.d1h`, `.c1h`), analog star (`.d2h`, `.c2h`), or analog sky (`.d3h`, `.c3h`).
GHRS	ACCUM	*n*	Each group is an independent subintegration with exposure time given by group parameter EXPOSURE. If FP-SPLIT mode was used, the groups will be shifted in wavelength space. The independent subintegrations should be coadded prior to analysis (see "Coadding Spectra" on page 55).
	RAPID	*n*	Each group is a separate subintegration with exposure time given by group parameter EXPOSURE.
WF/PC-1	WF	4	Group *n* represents CCD chip *n*, e.g., group 1 is chip 1 (unless all chips were not used). Group parameter DETECTOR always gives chip used.
	PC	4	Group *n* is chip *n* + 4, e.g., group 1 is chip 5. If all chips were not used, see the DETECTOR parameter which always gives the chip used.
WFPC2	All	4	Planetary chip is group 1, detector 1. Wide Field chips are groups 2–4 for detectors 2–4. If all chips were not used, see the DETECTOR keyword.

At this point it may be a good idea to verify that you received all the scientific data that you were expecting. We suggest you use the latest version of your Phase II proposal to verify that the tape contains all data you expected to receive.

The individual instrument chapters give examples for how to relate proposal exposure logsheet lines to data files.

CHAPTER 4: Getting Data From the Archive

In This Chapter...

STScI maintains an Archive of all HST data and associated calibration files. The Archive includes a large database describing the observations that have been made with the HST and the files contained in the Archive. An interface to the Archive, StarView, allows you to search the Archive and retrieve data from it.

In this chapter we will explain how to access and use the HST Data Archive. In the first section we describe how to get an account and access the Archive host computers at STScI. In the second section we work through a sample StarView session illustrating basic techniques, from login through retrieval and logout. The last section is a tutorial showing how to find and retrieve calibration reference files.

Getting Started Using the HST Archives

In this section we describe how to access the HST Archive through the host computers and how to register for a retrieval account.

Accessing the Archive Hosts

STScI has set up two Archive host computers for external access of the HST Archive: `stdata.stsci.edu` (VMS), and `stdatu.stsci.edu` (Unix). To get started using the HST Archive, external users should use telnet to connect to one of these host computers and log in with username `guest` and password `archive`.[1] Once you're in as guest, you can use the Archive user interface, StarView, to peruse the database. Simply type `starview` from the command line for the terminal version or `xstarview` for the X-Windows version.

You can also access the Archive computers via the World-Wide Web through the STScI Home Page (see "World Wide Web" on page 5).

To retrieve data, you'll need a registered account. The simplest way to register is through our World Wide Web form at the following URL:

`http://stdatu.stsci.edu/registration.html`

You may also register by typing the `register` command while logged onto one of the Archive host machines. This puts you in an editor on a registration form. After you have filled out the form, exit the editor, and the form will be mailed to the Archive support hotseat. Your retrieval account will be activated within two working days; you will get e-mail when it is ready. Once you have registered, you can use StarView to retrieve non-proprietary images and spectra.

General Observers and Guaranteed Time Observers normally have exclusive rights to their HST data for one-year. However, all observations obtained under calibration proposals are immediately non-proprietary.

Data retrieved through StarView will appear in the `data` directory of the Archive host computer. You will receive an e-mail message when the retrieval request has been queued, and another when it has been completed. You can then use anonymous FTP to transfer the files to your home computer.

While logged onto the Archive host computers, you can type `home` to return to your login directory, or `docs` to a directory of documents and manuals, including PostScript versions of the *HST Archive Manual* and the *HST Archive Primer*.

You can retrieve PostScript copies of the manuals using FTP (the FTP session will have to be started on your home machine, since neither `stdata` nor `stdatu` will allow outgoing FTP or telnet sessions). Alternatively, you can contact the Archive hotseat and ask for a hardcopy to be sent to you.

See the HST Archive's World Wide Web page for more information. If you have any questions, direct them via e-mail to `archive@stsci.edu`, or phone (410) 338-4547. The Archive web page is at URL:

`http://www.stsci.edu/archive.html`

1. European users should generally use the ST-ECF Archive system. Telnet to `stesis.hq.eso.org` and log in with the user name `starcat` or contact `catalog@eso.org` for help. Canadian users should request archival data through the CADC; contact `cadc@dao.nrc.ca` for help. The user interface at ST-ECF and CADC is STARCAT, not StarView.

A separate help desk (`archive@stsci.edu`) exists to provide technical assistance in using the HST Archive.

StarView Tutorial

In this section, a whole StarView session is presented, from logging in to the host computers through using FTP to move the data from the Archive host to your home computer. A second tutorial on selecting and retrieving calibration reference files is provided on page 471.

StarView is available in two forms:

- An X-Windows based version.
- A terminal version for basic terminals, such as a VT100.

In this manual we describe the X-Windows version but we also provide the information needed to run the terminal version.

The X-Windows version of StarView is available as distributed software which you install and run on your home computer. For information about how to install `xstarview` on your home system, contact the Archive hotseat. Alternatively, you could use telnet to login to the `stdatu` host using the guest account.

By running the `xstarview` client on your own computer, you avoid the overhead of running the software over the network, giving you a considerable speed improvement.

If you need to use telnet, here's how to do so.

```
% telnet stdatu.stsci.edu
Connected to stdatu.stsci.edu.
Escape character is '^]'.
Login: guest
Password: archive
```

The username and password for the guest account

After logging in (either by running the `xstarview` client on your computer or by using telnet to access `stdatu`), some introductory messages will appear on your screen. To see more of the text, press Space. To quit, press Q.

Start the X-Windows version of StarView by typing:

```
% xstarview
```

In the X-Windows version you will be asked for your X display host name. You should respond with the name of your home workstation.

You will then be instructed to add `stdatu` to your computer's `xhost` file by typing the following line in another window and then pressing Return to continue:

```
% xhost +stdatu.stsci.edu
```

If you want to use the terminal version instead, you would type:

```
% starview
```

In the terminal version, you will be asked to confirm your terminal setup. For example:

```
xterm 24 x 80 [Y]:
```

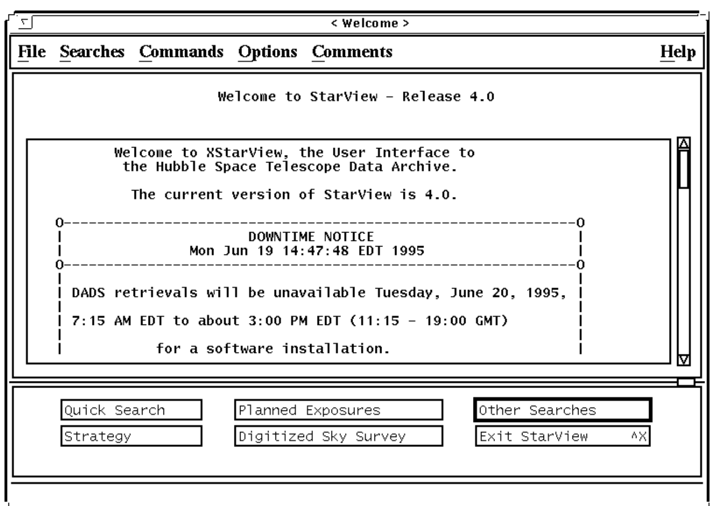
If this is correct, press Return to continue. If this is not correct, then answer "no" by pressing N followed by Return. If you answer "no", you will then be asked some questions about your terminal type, number of lines, and number of columns. Type a question mark (?) to get help about your options.

The StarView session is then started; messages will be displayed telling you what is happening (e.g., data dictionaries being loaded). This process may take a minute or two to complete.

Welcome Screen

The StarView <Welcome> screen (Figure 4.1) will appear (this and all subsequent screens are taken from the X-Windows version). If there is any urgent news (e.g., a message about possible system downtime), it will appear at the top of the welcome text.

Figure 4.1: Welcome Screen



You can scroll through the text and read any additional information below the display area by using the scroll bar on the X-windows version of StarView. On the terminal version (for VT100 or other basic terminals), use the arrow keys or page up by pressing Control-V and page down by pressing Control-P.

Command Usage and Screen Interaction

In the X-windows version of StarView:

- Use the mouse to select all functions.
- Choose options by positioning the mouse pointer over the command button or menu and pressing the left mouse button.

In the terminal version:

- Press Control-T to cycle through the three screen areas (*menu*, *work area*, and *command box*).
- Use the arrow keys to move around within any one portion of the screen.
- Whenever an option is highlighted, press Return to invoke the highlighted function.

You can also use the *command accelerators* to invoke functions (i.e., run commands). Some command buttons show accelerators such as “^N”, which means the function or command can be invoked by pressing down on the Control key while simultaneously pressing the N key. Other commands show accelerators such as “E+n”, which means that you would press the Esc (escape) key followed by N.

Searching the Catalog

To search the catalog:

1. Choose a search screen.
2. Specify your search criteria, such as position and a release date before today’s date so that you get public data.
3. Start the search by clicking on the **[Begin Search]** button.
4. Step through subsequent found observations looking for those of interest to you.

In this example we use the <Quick Search> screen to search the HST catalog.

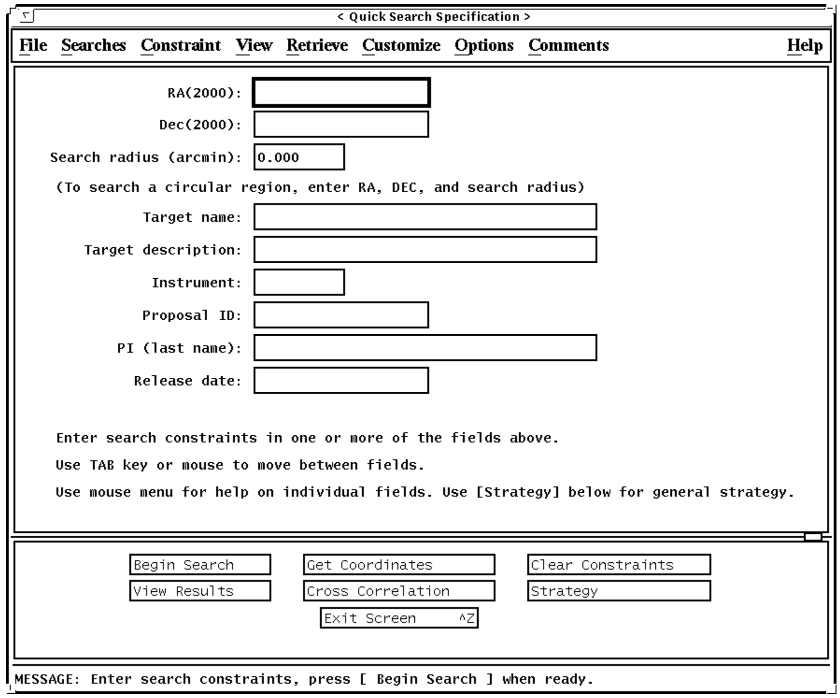
The <Quick Search> screen is useful for most basic searches of the HST catalog. An extensive set of more detailed search screens are also available. To choose one of these, click on **[Other Searches]** or pull down the **| Searches |** menu.

The <Quick Search> Screen

Choose the <Quick Search> screen by clicking the **[Quick Search]** button. (In `xstarview`, clicking is done by moving the cursor to the command button and pressing the left mouse button. In the terminal version, either (1) use Control-T to move the cursor to the command box, and the arrow keys to move the cursor until the button is highlighted then press Return, or (2) press Control-J, the accelerator for the <Quick Search> command.)

The <Quick Search> screen is shown in Figure 4.2. We will use this screen to search for all publicly available WFPC2 observations of M87.

Figure 4.2: Quick Search Screen



Specifying Search Criteria

There are various ways to search for observations of a particular target in the catalog. The easiest way is to enter the name (which you should embed in wildcard characters, e.g., `*mars*`) in the target field. Because observers do not necessarily use the same convention to name sources, this will typically not return all observations of a given source. The best way to be certain you retrieve *all* observations of a given target (for stationary targets) is to search for observations within a given (radial) distance of your source's position by entering constraints in the "RA", "Dec", and "Search radius" fields on the <Quick Search> screen.

If you do not know the RA and Dec of your target, you can run either the SIMBAD or NED target name resolver from within StarView. Each resolver automatically determines the target's position using a network connection to either the SIMBAD database in Europe or the NASA Extragalactic Database (NED) in California. It then populates the RA and Dec fields on the search screen with this information. Click on the **[Get Coordinates]** button to use the SIMBAD resolver; to use NED, pull down the **| Options |** menu, select User Defaults, and change the "Coordinates lookup server" field to NED.

In this case, we know the coordinates of M87, so we enter the RA and Dec for M87 and a search radius of 5 arcminutes in the corresponding fields on the search screen.

We want WFPC2 observations, so move to the "Instrument" field. The valid HST instruments are:

- Faint Object Camera (FOC).
- Faint Object Spectrograph (FOS).
- Fine Guidance Sensors (FGS).
- Goddard High Resolution Spectrograph (HRS).
- High Speed Photometer (HSP).
- Wide Field Planetary Camera (WFPC).
- Wide Field Planetary Camera 2 (WFPC2).

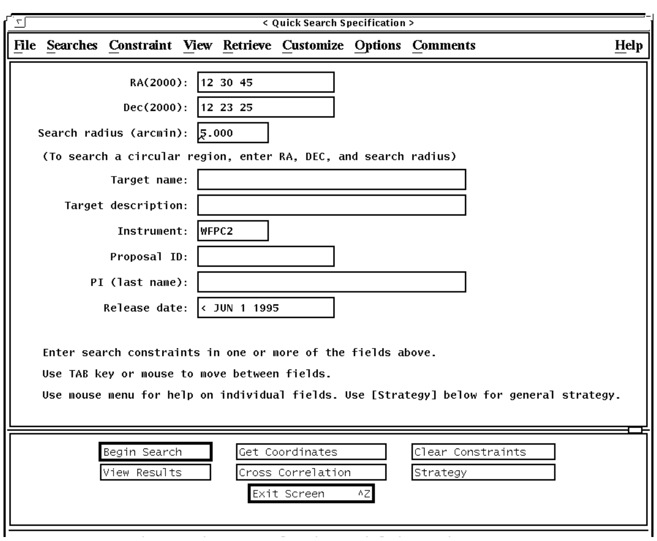
To get help on the valid ranges for any field, use the *field help*. In `xstarview`, move the cursor to the field and press the right mouse button (or press the Help button, often located in the bottom left corner of your keyboard). In the terminal version, move the cursor to the field and press Control-H.

Enter `WFPC2` in the instrument field. (To find observations from more than one instrument, use a comma-separated list; e.g., `WFPC2,WFPC,FOC`.)

We want public data, so now specify that we want data released prior to today's date. For example, move to the "Release date" field and enter `<Jun 1 1995` for any datasets that were released before June 1, 1995. Figure 4.3 shows how the <Quick Search> screen looks at this point.

Don't use commas in a date field. StarView will interpret the comma as a list operator.

Figure 4.3: Quick Search Screen With Constraints Entered



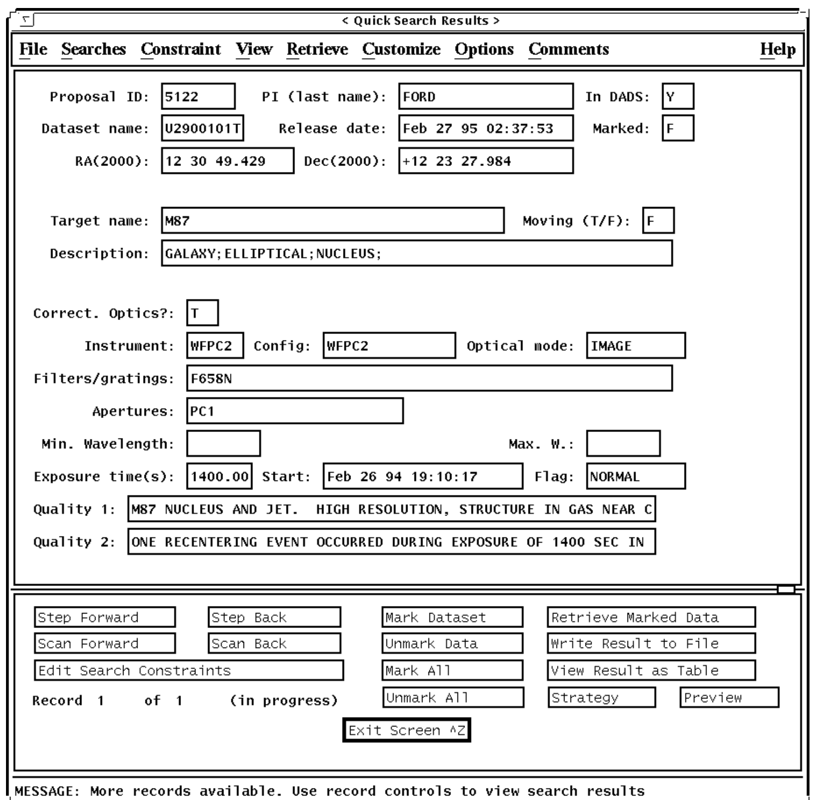
Use the **[Strategy]** button to get help using any StarView screen, or the pull down **| Help |** in the menu bar to see all the available StarView help.

Starting the Search

Click on the **[Begin Search]** button to search the catalog for the observations satisfying your search criteria. If none are found, a message will appear at the bottom of the screen, and you will need to enter different search constraints. If at least one observation is found, the screen will change to the <Quick Search Results> screen.

The <Quick Search Results> screen (Figure 4.4) shows the results of your catalog search. The first record that matches your search criteria will be displayed.

Figure 4.4: Quick Search Results Screen With Record Display



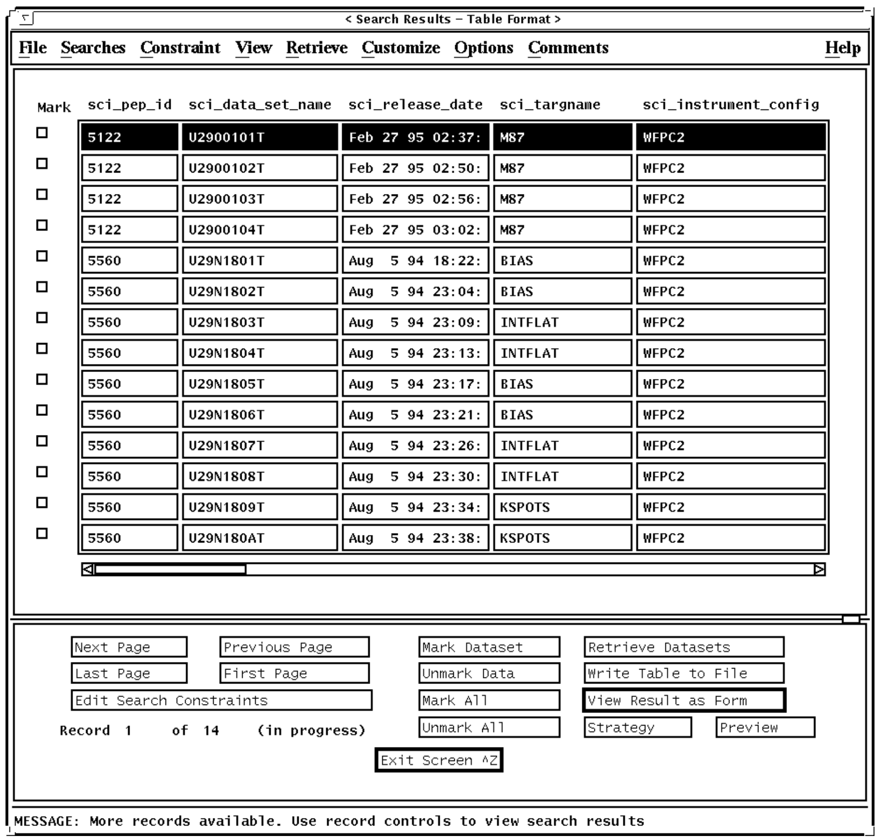
Viewing Subsequent Found Observations

If you want to scan the full list of your search results:

- Click the **[Step Forward]** button to view one record at a time.
- Click **[Scan Forward]** to see all of the found records in rapid succession. Press any key to stop the scan.
- To go back to previous records, use **[Step Back]** or **[Scan Back]**.

Another way to view your search results is to use StarView's <TableFormat> screen. Click on **[View Result as Table]** to see several catalog records at the same time (see Figure 4.5). Click on **[View Result as Form]** to return to the single-record screen format.

Figure 4.5: Quick Search Results Displayed on the Table Format Screen



Mark	sci_pep_id	sci_data_set_name	sci_release_date	sci_targname	sci_instrument_config
□	5122	U2900101T	Feb 27 95 02:37:	M87	WFPC2
□	5122	U2900102T	Feb 27 95 02:50:	M87	WFPC2
□	5122	U2900103T	Feb 27 95 02:56:	M87	WFPC2
□	5122	U2900104T	Feb 27 95 03:02:	M87	WFPC2
□	5560	U29N1801T	Aug 5 94 18:22:	BIAS	WFPC2
□	5560	U29N1802T	Aug 5 94 23:04:	BIAS	WFPC2
□	5560	U29N1803T	Aug 5 94 23:09:	INTFLAT	WFPC2
□	5560	U29N1804T	Aug 5 94 23:13:	INTFLAT	WFPC2
□	5560	U29N1805T	Aug 5 94 23:17:	BIAS	WFPC2
□	5560	U29N1806T	Aug 5 94 23:21:	BIAS	WFPC2
□	5560	U29N1807T	Aug 5 94 23:26:	INTFLAT	WFPC2
□	5560	U29N1808T	Aug 5 94 23:30:	INTFLAT	WFPC2
□	5560	U29N1809T	Aug 5 94 23:34:	KSPOTS	WFPC2
□	5560	U29N180AT	Aug 5 94 23:38:	KSPOTS	WFPC2

Use the **[Preview]** button to get a quick look at the data. This can help you decide whether or not to retrieve a dataset. Preview displays compressed HST images (not suitable for science analysis), as well as FOS or GHRS spectra. Only public data are available for preview. Preview is not available in the terminal version.

Retrieving Datasets From the Archive

We now want to retrieve some of the data that we have identified in the catalog. The steps in this process are:

1. Mark the observations that you want to retrieve; you can mark them either individually or as a group.

2. Display and review the list of datasets to be retrieved.

3. Specify the file formats and media to be used in the retrieval process.
4. Submit the request.
5. Check the request status, if desired.

Marking Observations for Retrieval

To mark for retrieval the dataset displayed on the screen, click the **[Mark Dataset]** button. This action will be confirmed by a message at the bottom of the screen. Also, the “Marked” field, in the upper right corner of the screen, will display “T” (true) indicating that the dataset has been marked for retrieval.

You can mark datasets for retrieval in either the table-row format display screen, in which case the highlighted record is marked, or on the <Quick Search Results> screen with the record displayed.

If you want to mark for retrieval *all* of the records matching your search criteria, click on the **[Mark All]** button. This could be a large volume of data, and it would be for the M87 search request described here. Alternatively, step through your search results records by clicking on the **[Step Forward]** button and only click on the **[Mark Dataset]** button for a few of the observations.

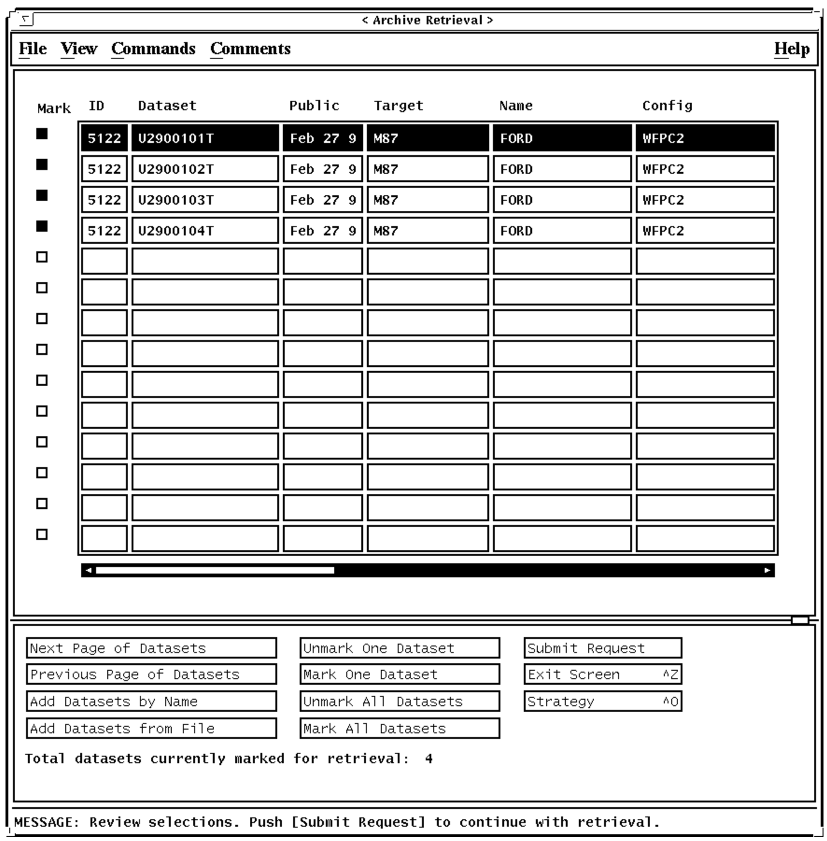
Reviewing the Retrieval Request

Once you have marked records for retrieval, you begin the retrieval process by displaying and reviewing a list of datasets to be retrieved. To do this:

1. Click on the **[Retrieve Marked Data]** button to exit the <Quick Search Results> screen and to begin the retrieval process by bringing up the <Archive Retrieval> screen.
2. Review the list of datasets.

The <Archive Retrieval> screen lists all of the datasets that you have marked for retrieval. In this case, you would see something like Figure 4.6.

Figure 4.6: Archive Retrieval Screen



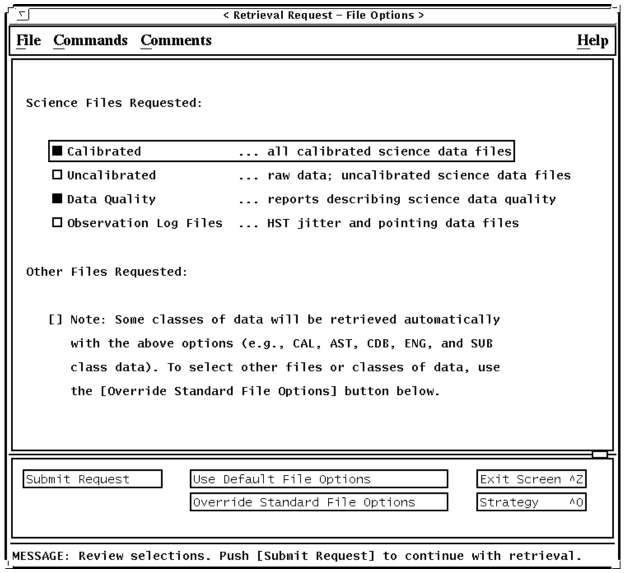
Review the list of datasets that you have marked for retrieval. If you have marked several datasets, you may need to click on the **[Next Page of Datasets]** button to see additional screens of marked records. The total number of datasets that you have marked for retrieval is shown near the bottom of the screen.

Specifying Formats and Media

1. Continue with the data retrieval process by clicking the **[Submit Request]** button.
2. Specify the files that you want to retrieve.
3. Specify the type of media (file transfer method) that you want.

When you click the **[Submit Request]** button, the <Retrieval Request - File Options> screen is displayed (Figure 4.7).

Figure 4.7: Retrieval Request - File Options Screen

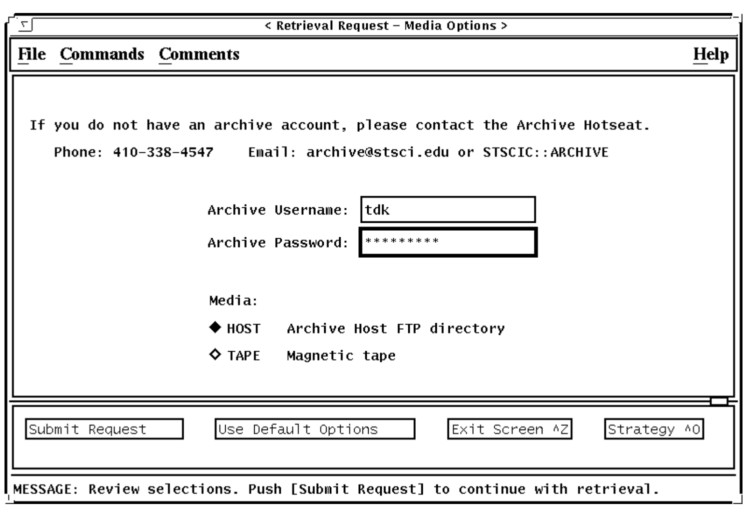


The <Retrieval Request - File Options> screen indicates the kinds of files that will be retrieved, in this case the final calibrated science data files and the data quality report files will be retrieved. These defaults are acceptable. Click the **[Submit Request]** button to continue with the retrieval process.

The <Retrieval Request - Media Options> screen is then displayed (Figure 4.8). You will need to enter your Archive user name and password, pressing Return after each entry.

If you do not have an Archive account, then you will not be able to retrieve data until you have registered with STScI. For information about how to register, see “Accessing the Archive Hosts” on page 89.

Figure 4.8: Retrieval Request- Media Options Screen



The screen will indicate HOST (retrieve to Archive host staging disk) as the default distribution medium. This is acceptable—your data will be retrieved to a subdirectory of the data disk on `stdatu`. You will then be able to FTP it electronically from there to your home computer, as described in "Getting Your Data" on page 103.

You may also use this screen to specify that your data be put onto a tape and mailed to you. This option is especially useful for large data requests.

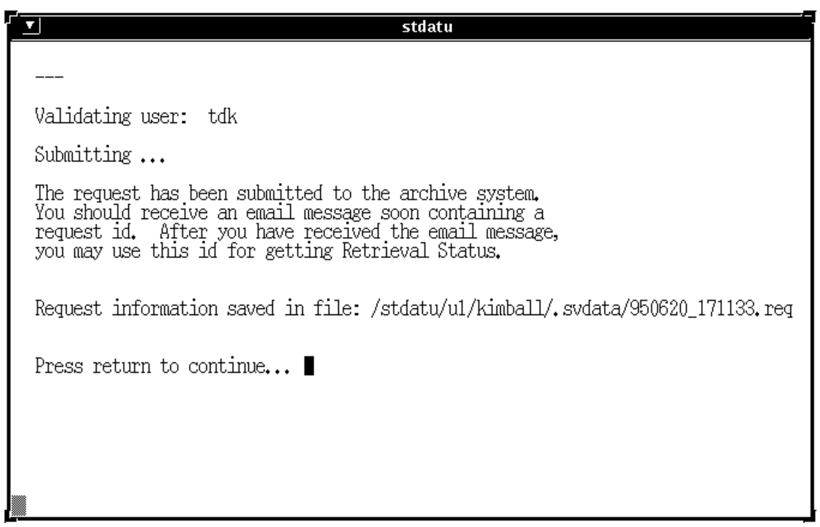
Submit the Request

Click on the **[Submit Request]** button to begin the submission process. StarView will validate your Archive account information and send your retrieve request to the Archive system.

At this point `xstarview` will want to interact with you using a special xterm window which it will start up. Look at that window and respond to any `xstarview` requests that appear there.

The list of datasets you have requested will be saved in a file named after the date and time of the request, with an extension of `.req`. The name of this file will be displayed in the xterm window. Figure 4.9 shows how a StarView screen might look at this stage.

Figure 4.9: Retrieval System Messages.



Press Return to exit from the retrieval process and to return to the StarView screen from which you initiated the retrieval request.

Shortly after your request has been submitted, you will receive an e-mail message telling you that your request has been accepted and queued by the Archive system, and will also give you the request ID.

You can use the request ID later to check the status of your request, and also to locate your data on the Archive host's staging disk after it has been retrieved.

Checking Request Status

You can check the status of your retrieval request, to do this:

1. Click on the **[Retrieval Status]** button from within the **| Retrieve |** menu on most StarView search screens, or click on the **| Commands |** menu from the <Welcome> screen.
2. You will be asked to enter your request ID (this will be e-mailed to you shortly after your request is submitted). Type the request ID.
3. Press Return to continue with your StarView session. Figure 4.10 shows a sample retrieval status screen.

Figure 4.10: Sample Retrieval Status Screen

```
StarView xterm

Enter request id or user id [kimb]: tdk7992

       Status of the latest request submitted:  tdk7992
       ---------------------------------------

 Submitted:   Jun 20 1995 10:09:33:440PM  (GMT)
 Completed:   Jun 20 1995 10:57:50:463PM

 Status  MBytes      Number of Files
 ------  ----------  ---------------
 S           0.0388               12

 I : Files waiting to be processed.
 S : Files retrieved successfully.

 D : Files not retrieved: problem with destination directory.
 F : Files not retrieved: FTP error.
 G : Files not retrieved: invalid generation date.
 H : Files not retrieved: unknown host.
 N : Files not retrieved: dataset does not exist.
 P : Files not retrieved: proprietary restriction.
 T : Files not retrieved: transfer error.
 V : Files not retrieved: invalid version number.
```

Exiting StarView

You can now either continue working in StarView, or you can exit. You could also logout of the Archive host altogether and wait for the mail message that will tell you that the files have been retrieved and are ready for you. This message will go to the e-mail account that you identified when you registered for an Archive account.

Press Control-X to exit StarView. A dialog box will appear asking you to confirm that you really want to exit. Click **[OK]** to exit.

Getting Your Data

After your data have been retrieved from the Archive, you will receive a second e-mail notification. You may then transfer the data back to your home site via anonymous FTP. Figure 4.11 shows a sample FTP session.

Figure 4.11: Retrieving Files Using FTP

```
% ftp stdatu.stsci.edu
Connected to stdatu.stsci.edu.
220 stdatu.stsci.edu FTP server (Version 5.86) ready.
Name (stdatu.stsci.edu): anonymous
Password: Type your e-mail address
.
<message of the day displayed here>
.
ftp> cd tdk7992
250 CWD command successful.
ftp> binary
200 Type set to I.
ftp> prompt
Interactive mode off.
ftp> mget u*.fit
200 PORT command successful.
50 Opening BINARY mode data connection for u2900101t_c0f.fit
226 Transfer complete
.
.
ftp> bye
221 Goodbye.
```

Don’t forget to set the FTP transfer type to “binary” before transferring the files.

You are now ready to begin analyzing your dataset. Note that the data are in FITS format and may be converted to STSDAS format using the task **strfits**.

Remember, if you have any problems or questions, contact the Archive hotseat at `archive@stsci.edu`.

Tutorial: Retrieving Calibration Reference Files

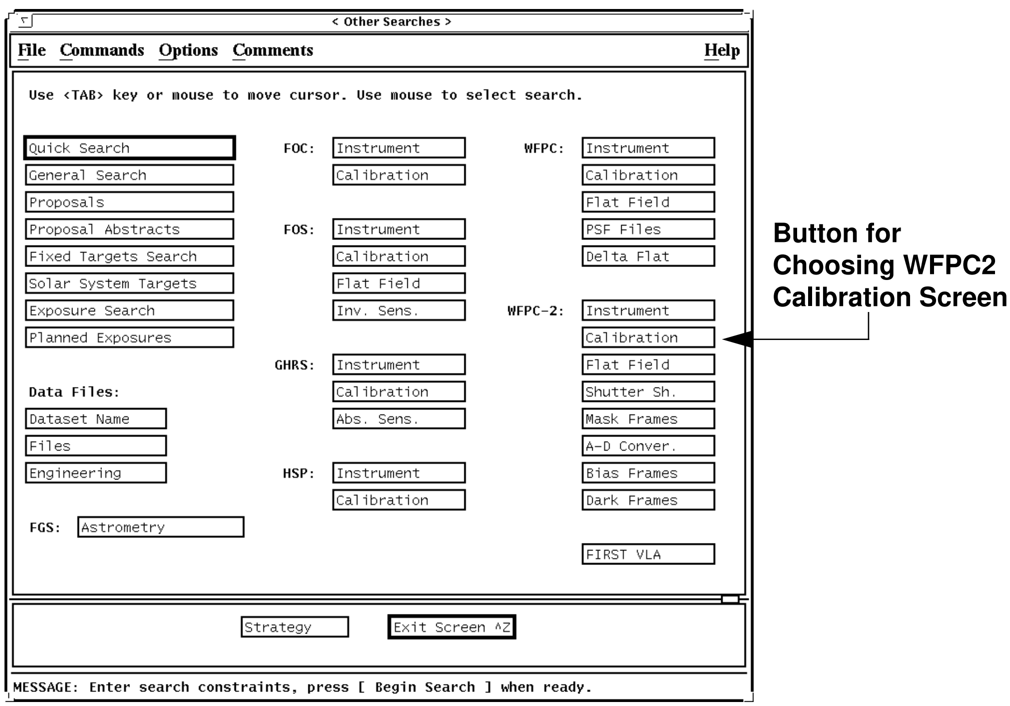
StarView provides calibration reference file screens for each instrument. These screens let you see which calibration files and tables were used to calibrate a given dataset by the PODPS pipeline, and which files and tables are currently recommended. You can mark either the used or the recommended reference files and tables for retrieval and retrieve them through StarView.

If you already know the name of the calibration reference file or table you wish to retrieve (e.g., if you have determined the file name from a listing on STEIS), then you can retrieve just that file by using the **[Add Datasets by Name]** function on the <Archive Retrieval> screen. This is described on page 106.

In this example, we use a StarView reference file screen to retrieve both the “used” and “recommended” calibration files for an M87 dataset.

1. Start StarView as described in “StarView Tutorial” on page 91. This will bring up the <Welcome> screen (Figure 4.3).
2. Click the **[Other Searches]** button.[2] (We want to find the search screen for calibration reference files.) The **[Other Searches]** screen will be displayed (Figure 4.12), with the cursor highlighting “Quick Search”.

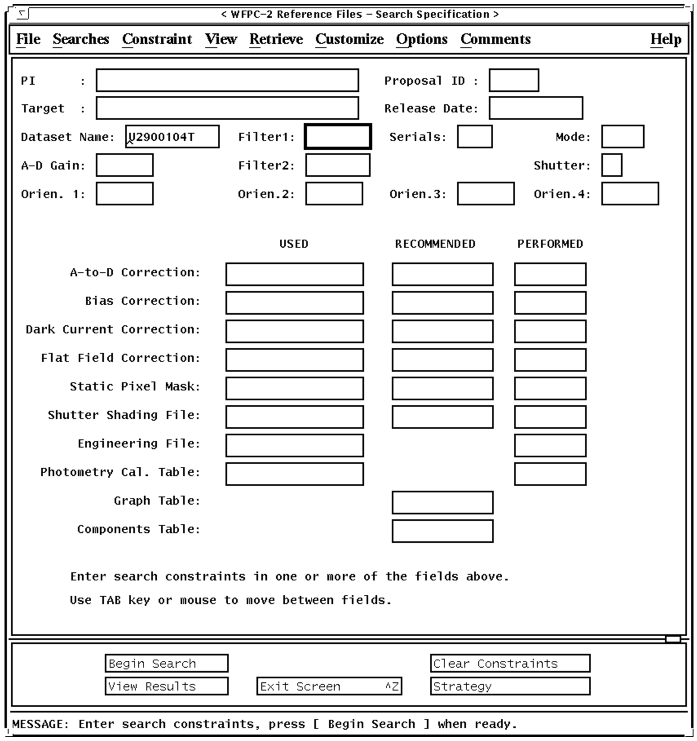
Figure 4.12: Other Searches Screen



3. Select the <WFPC2 Reference Files> screen because we will retrieve calibration files for a WFPC2 observation. The <WFPC2 Reference Files - Search Specification> screen will be displayed.
4. Specify criteria. We want to specify a particular dataset, so press the arrow keys until the cursor moves to the “Rootname” field, and then type the dataset name for the observation whose calibration files will be retrieved. For example, enter `U2900101T` (see Figure 4.13).

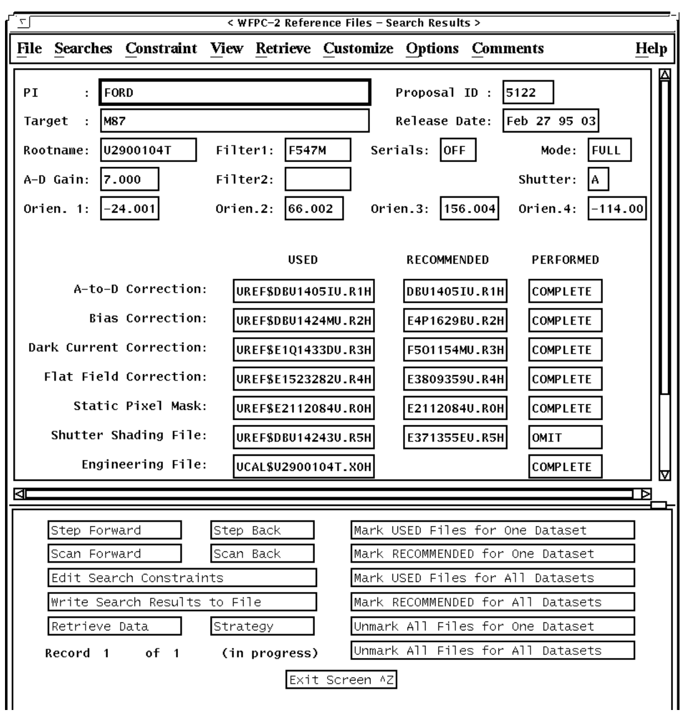
2. Users of the terminal version can use the accelerator E+2.

Figure 4.13: WFPC Reference Files - Search Specification Screen (Constrained)



5. Click on the **[Begin Search]** button to submit your catalog search request. The <WFPC2 Reference Files - Search Results> screen will be displayed for the observation that you specified (Figure 4.14).

Figure 4.14: WFPC2 Reference Files - Search Results Screen



6. Click **[Mark USED Files for One Dataset]** to mark for retrieval those calibration files actually used to calibrate the dataset. If the files listed in the RECOMMENDED column differ from those in the USED column, then you can click on **[Mark RECOMMENDED for One Dataset]** to retrieve the calibration files that are now recommended for calibrating the data.
7. Click the **[Retrieve Marked Data]** button to begin the retrieval process for the marked reference files. Continue with the data retrieval procedures as outlined in "Retrieving Datasets From the Archive" on page 97.

The defaults on the <Retrieval Request-File Option> screen, i.e., "Calibrated", will return the correct files for the specified calibration reference files and tables.

Retrieving a File By Name

If you know the name of a file or dataset that you wish to retrieve from the Archives, you can retrieve it directly using the **[Add Datasets by Name]** button from the <Archive Retrieval> screen. To get to this screen from the <Welcome> screen, pull down the **| Commands |** menu in the menu bar (see Figure 4.1) and choose the **[Retrieve Marked Datasets]** option. This will place you in the <Archive Retrieval> screen.

Choose the **[Add Datasets by Name]** command from the <Archive Retrieval> screen (or use **[Add Datasets from File]** if you have a list of dataset names). Enter the rootname (*no extension*) of the calibration reference file or science file you wish to retrieve. When you have added all the rootnames for the files you wish to retrieve, click **[Submit Request]** and proceed with the retrieval process as described on page 97.

CHAPTER 5: Observation Logs

In This Chapter...

This chapter describes the new *Observation Log Files* that are available to General Observers. These files can be used to obtain pointing and jitter data for any HST observation. Instructions for retrieving the files from the HST Archive via StarView and examples of how to use STSDAS to extract meaningful information from the files are provided.

What are Observation Log Files?

A set of pointing and specialized engineering data, called *observation logs*, are now being provided to General Observers. These data are produced by the Observatory Monitoring System (OMS), an automated software system that interrogates the HST engineering telemetry and correlates the time-tagged engineering stream with the scheduled events as determined from the Mission Schedule, the seven-day command and event list that drives all HST activities. The Observatory Monitoring System reports the status of the instruments and observatory and flags discrepancies between planned and executed actions. Each log contains the engineering data in a time period encompassing the science exposure. The data are in FITS format and may be converted to STSDAS binary table format using the STSDAS **strfits** task and manipulated using the STSDAS table tools (**ttools**). The observation logs are named according to the first eight characters of the root name (plus the letter “J”).

The observation log contains a variety of interesting information and are necessary in evaluating and reconstructing pointing stability and environment during a given exposure. The files are intended for General Observers, STScI scientists, and engineers.

OMS provides observers (and archival researchers) with observational information about guide star acquisition, pointing, and tracking that is not normally provided in the science headers. OMS was installed in operations during October 1994, at which time the `.cmh`, `.cmj`, `.cmi` files were generated. Pointing and tracking information prior to October 1994 is not routinely available to observers but can be requested by sending e-mail to `help@stsci.edu`. OMS observation logs changed to the `.jih`/`.jid` image format after August 1995. The `.cmi` table continued to be generated, but is renamed `.jit` to keep the naming convention consistent with the new software. A more in-depth description of OMS can be found in *Observation Logs*, by O. Lupie and B. Toth. This document can be requested via e-mail to `help@stsci.edu.`

In the OMS version of August 1995, `.cmj` tables were replaced with a jitter image, which is a two-dimensional histogram of jitter excursions during the observation. The jitter image has an extension of `.jih` for the header and `.jid` for the image data. The `.cmi` table accompanies the jitter image. The header file of the image replaces the `.cmh` file but includes the same information with the addition of some image-related keywords.

Contents of Headers, Tables, and Jitter Images

The OMS data header file (`.cmh` or `.jih`) contains information about the observation. The header is divided into groups of keywords that deal with a particular topic (i.e., SPACECRAFT DATA, BACKGROUND LIGHT, POINTING CONTROL DATA, and LINE OF SIGHT JITTER SUMMARY). A short description of each keyword is provided in the header. Table 5.1 lists the OMS data file extensions and the corresponding file contents.

Table 5.1: Observation Management System (OMS) Observation Logs

Extension	Contents
	October 1994 to August 1995
`.cmh`	OMS header
`.cmj`	High time resolution (IRAF table)
`.cmi`	Three-second averages (IRAF table)
	After August 1995
`.jih/.jid`	Two-dimensional histogram (image)
`.jit`	Three-second averages (IRAF table)

In the following section we give brief descriptions of the contents of these headers and tables.

rootname`j.cmh`: The ASCII header file contains the time interval, the rootname, averages of the pointing and spacecraft jitter, the guiding mode, guide star information, and alert or failure keywords. Figure 5.1 is a representative header. Note that this header is similar to the headers attached to the `.cmi` tables with the exception that the ASCII file header (`.cmh`) contains problem flags and status warnings.

A header associated with the jitter image replaces the `.cmh` header in a later version of OMS. It does, however, contain the same information and keywords.

rootname`j.cmj`: This table presents the data at the highest time resolution for the telemetry mode in use. It contains the reconstructed pointing, guide star coordinates, derived jitter at the instrument aperture, and pertinent guiding-related flags. The intent is twofold: (1) to provide high-time resolution jitter data for deconvolution calculations or for assessing small aperture pointing stability, and (2) to display the slew and tracking anomaly flags with the highest resolution. Table 5.2 lists the table column heading, units and a brief definition.

Table 5.2: Contents of `.cmj` Table

Parameter	Units	Description
`seconds`	seconds	Time since window start
`V2 dom`	arcseconds	Dominant FGS V2 coordinate
`V3 dom`	arcseconds	Dominant FGS V3 coordinate
`V2 roll`	arcseconds	Roll FGS V2 coordinate
`V3 roll`	arcseconds	Roll FGS V3 coordinate
`SI V2`	arcseconds	Jitter at aperture reference
`SI V3`	arcseconds	Jitter at aperture reference
`RA`	degrees	Right ascension of aperture reference
`DEC`	degrees	Declination of aperture reference
`Roll`	degrees	Angle between North and +V3
`DayNight`	0,1 flag	Day (0) or night (1)
`Recenter`	0,1 flag	Recentering status
`TakeData`	0,1 flag	Vehicle guiding status
`SlewFlag`	0,1 flag	Vehicle slewing status

Figure 5.1: A Representative `.cmh` or `.jih` Header

```
--------------------------------------------------------------------------------

SIMPLE  =                    F / data conforms to FITS standard               !
BITPIX  =                   32 / bits per data value                          !
DATATYPE= 'INTEGER*4         ' / datatype of the group array                  !
NAXIS   =                    2 / number of data axes                          !
NAXIS1  =                   64 / length of the 1st data axis                  !
NAXIS2  =                   64 / length of the 2nd data axis                  !
GROUPS  =                    T / image is in group format                     !
GCOUNT  =                    1 / number of groups                             !
PCOUNT  =                    0 / number of parameters                         !
PSIZE   =                    0 / bits in the parameter block                  !
OMS_VER = '16.2C             ' / OMS version used to process this observation
PROCTIME= '1994.133:06:24:18.35' / date-time OMS processed observation
                                / date-times format (yyyy.ddd:hh:mm:ss.ss)

                               / IMAGE PARAMETERS
CRVAL1  =                  0.0 / right ascension of zero-jitter pixel (deg)
CRVAL2  =                  0.0 / declination of zero-jitter pixel (deg)
CRPIX1  =                   32 / x-coordinate of zero-jitter pixel
CRPIX2  =                   32 / y-coordinate of zero-jitter pixel
CTYPE1  = 'RA---TAN          ' / first coordinate type
CTYPE2  = 'DEC--TAN          ' / second coordinate type
CD1_1   =                  0.0 / partial of ra w.r.t. x (deg/pixel)
CD1_2   =                  0.0 / partial of ra w.r.t. y (deg/pixel)
CD2_1   =                  0.0 / partial of dec w.r.t. x (deg/pixel)
CD2_2   =                  0.0 / partial of dec w.r.t. y (deg/pixel)
COORDSYS= 'WFPC2             ' / image coordinate system
XPIXINC =                  2.0 / plate scale along x (mas per pixel)
YPIXINC =                  2.0 / plate scale along y (mas per pixel)
PARITY  =                   -1 / parity between V2V3 frame and image frame
BETA1   =               134.72 / angle from +V3 to image +x (toward +V2)
BETA2   =               224.72 / angle from +V3 to image +y (toward +V2)

                               / OBSERVATION DATA
PROPOSID=                05233 / PEP proposal identifier
PROGRMID= '288               ' / program id (base 36)
OBSET_ID= '02                ' / observation set id
OBSERVTN= '03                ' / observation number (base 36)
TARGNAME= 'NGC3379-PO        ' / proposer's target name
STARTIME= '1994.133:06:24:18.35' / predicted observation window start time
ENDTIME = '1994.133:06:39:18.35' / predicted observation window end time
SOGSID  = 'U2880203          ' / SOGS observation name                        !

                               / SCIENTIFIC INSTRUMENT DATA
CONFIG  = 'WFPC2             ' / proposed instrument configuration
PRIMARY = 'SINGLE            ' / single, parallel-primary, parallel-secondary
OPERATE = '1994.133:06:22:46.91' / predicted time instr. entered operate mode
TLMFORM = 'PN                ' / telemetry format
APERTURE= 'UWFALL            ' / aperture name
APER_V2 =                1.565 / V2 aperture position in vehicle frame (arcsec)
APER_V3 =                7.534 / V3 aperture position in vehicle frame (arcsec)

                               / SPACECRAFT DATA
ALTITUDE=               593.23 / average altitude during observation (km)
LOS_SUN =               106.08 / minimum line of sight to Sun (deg)
LOS_MOON=                77.11 / minimum line of sight to Moon (deg)
SHADOENT= '1994.133:05:11:29.00' / predicted Earth shadow last entry
SHADOEXT= '1994.133:05:42:45.00' / predicted Earth shadow last exit
LOS_SCV =                12.46 / minimum line of sight to S/C veloc. (deg)
LOS_LIMB=                 58.0 / average line of sight to Earth limb (deg)


                               / BACKGROUND LIGHT
ZODMOD  =                 22.3 / zodiacal light - model (V mag/arcsec2)
EARTHMOD=                 20.2 / peak Earth stray light - model (V mag/arcsec2)
MOONMOD =                 35.5 / moon stray light - model (V mag/arcsec2)
GALACTIC=                 -1.0 / diffuse galactic light - model (V mag/arcsec2)

                               / POINTING CONTROL DATA
GUIDECMD= 'FINE LOCK         ' / commanded guiding mode
GUIDEACT= 'FINE LOCK         ' / actual guiding mode at end of GS acquisition
GSD_ID  = '0084900235        ' / dominant guide star id
GSD_RA  =            161.70720 / dominant guide star RA (deg)
GSD_DEC =             12.45407 / dominant guide star DEC (deg)
```

Figure 5.1 (continued): Representative .cmh or .jih Header

```
GSD_MAG =               12.867 / dominant guide star magnitude
GSR_ID  = ‘0085201189        ‘ / roll guide star id
GSR_RA  =            161.93314 / roll guide star RA (deg)
GSR_DEC =             12.78141 / roll guide star DEC (deg)
GSR_MAG =               12.977 / roll guide star magnitude
GSACQ   = ‘1994.133:06:31:02.92’ / actual time of GS acquisition completion
PREDGSEP=             1420.775 / predicted guide star separation (arcsec)
ACTGSSEP=             1421.135 / actual guide star separation (arcsec)
GSSEPRMS=                  3.8 / RMS of guide star separation (milli-arcsec)
NLOSSES =                    0 / number of loss of lock events
LOCKLOSS=                  0.0 / total loss of lock time (sec)
NRECENT =                    0 / number of recentering events
RECENTR =                  0.0 / total recentering time (sec)

                               / LINE OF SIGHT JITTER SUMMARY
V2_RMS  =                  4.5 / V2 axis RMS (milli-arcsec)
V2_P2P  =                 51.6 / V2 axis peak to peak (milli-arcsec)
V3_RMS  =                 20.9 / V3 axis RMS (milli-arcsec)
V3_P2P  =                267.3 / V3 axis peak to peak (milli-arcsec)
RA_AVG  =            161.85226 / average RA (deg)
DEC_AVG =             12.58265 / average dec (deg)
ROLL_AVG=            293.01558 / average roll (deg)

                               / PROBLEM FLAGS, WARNINGS and STATUS MESSAGES
                               / (present only if problem exists)
ACQ2FAIL= ‘                 T’ / target acquisition failure
GSFAIL  = ‘DEGRADED          ‘ / guide star acquisition failure  (*1)
TAPEDROP= ‘                 T’ / possible loss of science data
TLM_PROB= ‘                  ‘ / problem with the engineering telemetry
TM_GAP  =               404.60 / duration of missing telemetry (sec)
SLEWING = ‘                 T’ / slewing occurred during this observation
TAKEDATA= ‘                 F’ / take data flag NOT on throughout observation
SIPROBnn= ‘                  ‘ / problem with specified science instrument (*2)

END
-----------------------------------------------------------------------------

notes
*1 - GSFAIL appears only once in a single header file.
     The following table lists all current possible values for the GSFAIL
     keyword:
     -------------------------------------------------------------
     GSFAIL    |DEGRADED          | / guide star acquisition failure
               |IN PROGR          | / guide star acquisition failure
               |SSLEXP            | / guide star acquisition failure
               |SSLEXS            | / guide star acquisition failure
               |NOLOCK            | / guide star acquisition failure
               |                  |
               |SREXCS?           | / guide star acquisition failure
               |SREXCS1           | / guide star acquisition failure
               |SREXCS2           | / guide star acquisition failure
               |SREXCS3           | / guide star acquisition failure
               |                  |
               |SREXCP?           | / guide star acquisition failure
               |SREXCP1           | / guide star acquisition failure
               |SREXCP2           | / guide star acquisition failure
               |SREXCP3           | / guide star acquisition failure
               |                  |
               |UNKNOWN           | / guide star acquisition failure
               |VEHSAFE           | / guide star acquisition failure
     -------------------------------------------------------------

*2 - The SIPROBnn keywords appear in the header file with nn = 01 - 99.
     The following table lists all current possible values for the SIPROBnn
     keyword:
     -------------------------------------------------------------------------
     SIPROBnn  |DCF_NUM unchanged| / This observation may not have been taken
               |FOS Safing!       | / This observation affected when FOS Safed!
               |HRS Safing!       | / This observation affected when HRS Safed!
               |WFII Safing!      | / This observation affected when WFII Safed!
               |FOC Safing!       | / This observation affected when FOC Safed!
               |Shut              | / FOS aperture door is not Open!
               |FAILED            | / FGS astrometry target acquisition failed
     -------------------------------------------------------------------------
```

rootname`j.cmi`: This table contains data that were averaged over three-second intervals. It includes the same information as the `.cmj` table and also includes orbital data (e.g., latitude, longitude, limb angle, magnetic field values, etc.) and instrument-specific items. This table is best suited for quick-look assessment of pointing stability and for studying trends in telescope or instrument performance with orbital environment. Table 5.3 lists the table column heading, units and a brief definition.

Table 5.3: Contents of `.cmi` Table, Three-Second Averaging

Parameter	Units	Description
`seconds`	seconds	Time since window start
`V2 dom`	arcseconds	Dominant FGS V2 coordinate
`V3 dom`	arcseconds	Dominant FGS V3 coordinate
`V2 roll`	arcseconds	Roll FGS V2 coordinate
`V3 roll`	arcseconds	Roll FGS V3 coordinate
`SI V2 AVG`	arcseconds	Mean jitter in 3 seconds
`SI V2 RMS`	arcseconds	rms jitter in 3 seconds
`SI V2 P2P`	arcseconds	Peak jitter in 3 seconds
`SI V3 AVG`	arcseconds	Mean jitter in 3 seconds
`SI V3 RMS`	arcseconds	rms jitter in 3 seconds
`SI V3 P2P`	arcseconds	Peak jitter in 3 seconds
`RA`	degrees	Right ascension of aperture reference
`DEC`	degrees	Declination of aperture reference
`Roll`	degrees	Angle between North and +V3
`LimbAng`	degrees	Angle between earth limb and target
`TermAng`	degrees	Angle between terminator and target
`LOS_Zenith`	degrees	Angle between HST zenith and target
`Latitude`	degrees	HST subpoint latitude
`Longitude`	degrees	HST subpoint longitude
`Mag V1,V2,V3`	degrees	Magnetic field along V1, V2, V3
`EarthMod`	V Mag/arcsec^2	Model earth background light
`SI_Specific`	–	Special science instrument data
`DayNight`	0,1 flag	Day (0) or night (1)
`Recenter`	0,1 flag	Recentering status
`TakeData`	0,1 flag	Vehicle guiding status
`SlewFlag`	0,1 flag	Vehicle slewing status

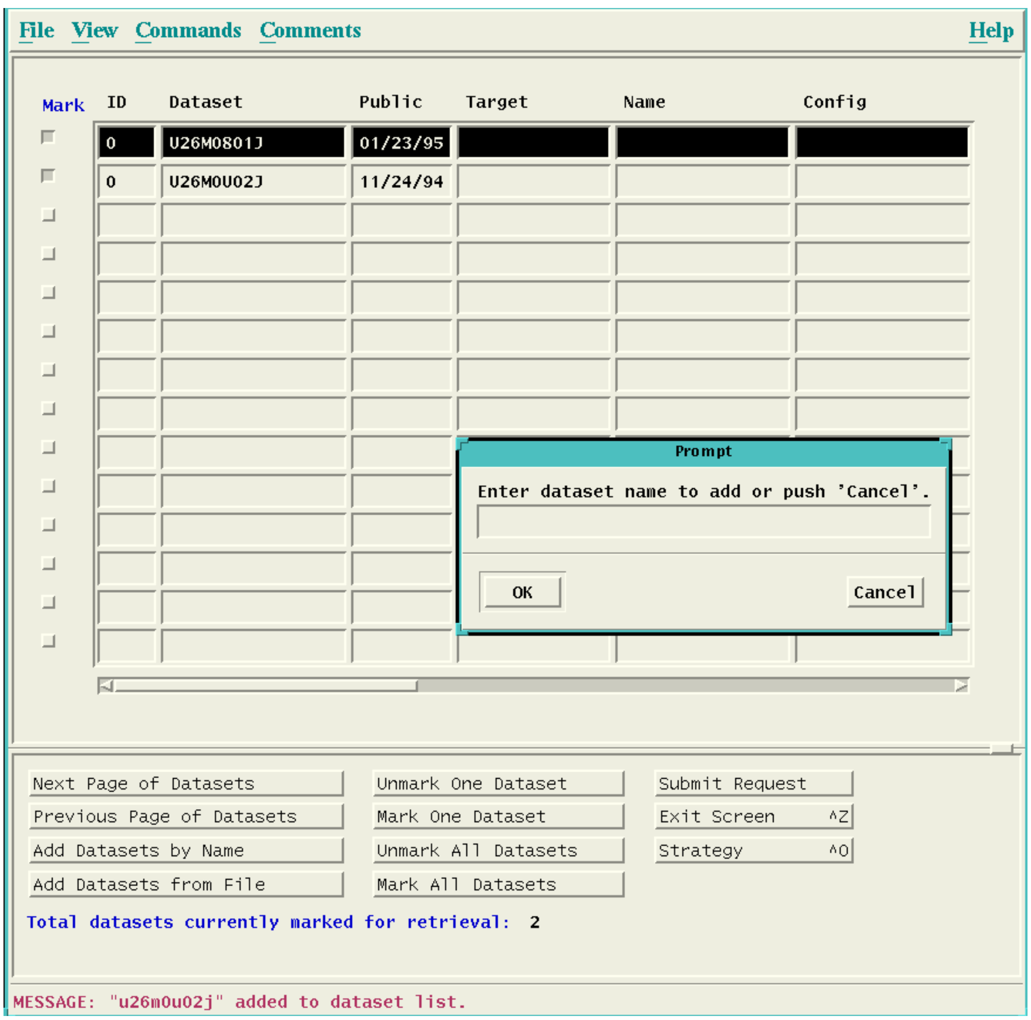
How to Access Observation Log Files

For data taken after October 20, 1994, observation log files can be retrieved from the HST Archive. Unlike science data, which generally has a one-year proprietary period, observation log files become public as soon as they are archived.

To access the observation log files, you will use StarView to retrieve the files from the Archive:

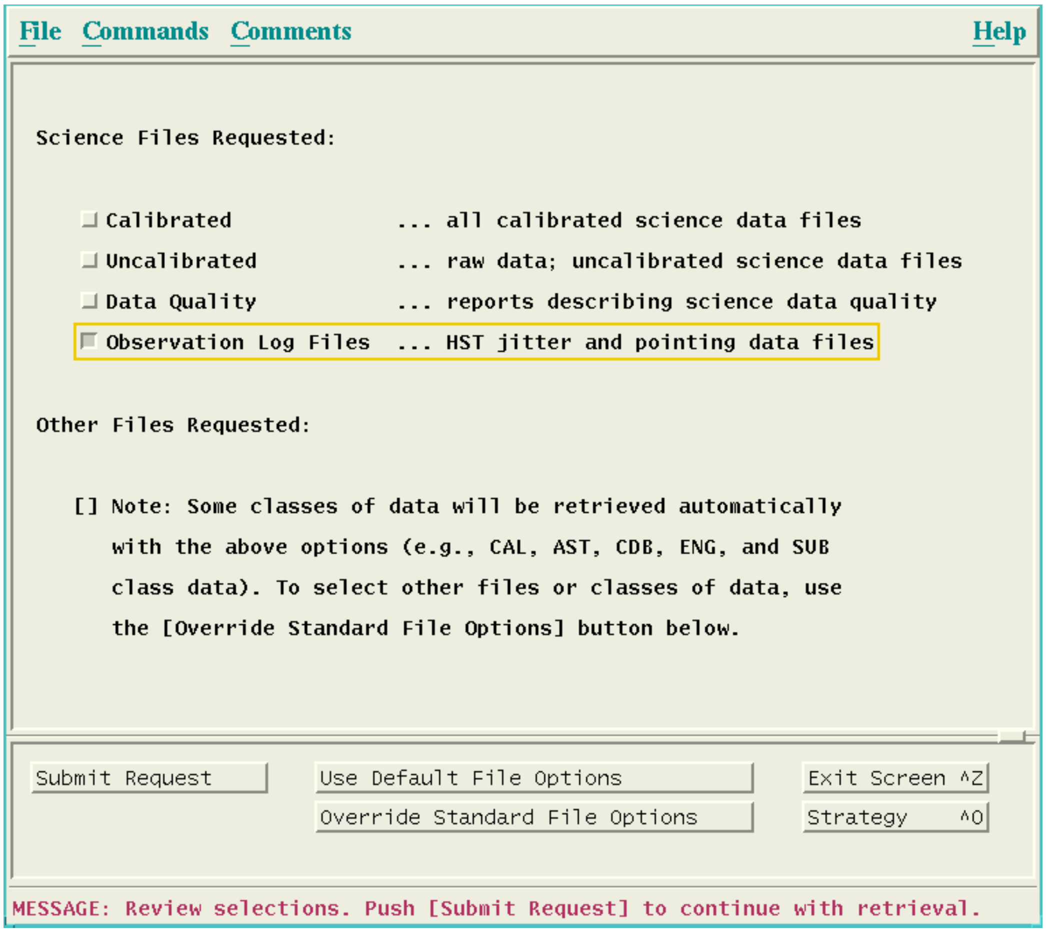
1. Enter the rootnames of the observation log files that you want on StarView's <Retrieve Marked Datasets> screen (the names can be entered individually, or read from a file on your disk). The rootname will have the same first 8 characters as the science data, but will be followed by the letter "J" instead of the usual "T". For example, if the science rootname is `U26M0801T`, then the observation log rootname will be `U26M0801J`.
2. Click on **[Submit Request]**.

Figure 5.2: Retrieving Observation Log Files in StarView



3. Now, on the <File Options> Screen, select “Observation Log Files”, as shown in Figure 5.3. Again, click on **[Submit Request]**, then enter your archive username and password, and the destination information.

Figure 5.3: Choosing Observation Log Files in StarView



How to Use Observation Log Files

Here are some simple examples of what can be learned from the observation log files.

Guiding Mode

Unless requested, all observations will be scheduled with FINE LOCK guiding, which may be one or two guide stars (dominant and roll). The spacecraft may roll slightly during an observation if only one guide star is acquired. The amount of roll depends upon the gyro drift at the time of the observation, the location during an orbit, and the lever arm from the guide star to the center of the aperture.

There are three commanded guiding modes: FINE LOCK, FINE LOCK/GYRO, and GYRO. OMS header keywords GUIDECMD (commanded guiding mode) and GUIDEACT (actual guiding mode) will usually agree. If there was a problem, they won't agree and the GUIDEACT value will be the guiding method actually used during the exposure. If the acquisition of the second guide star fails, the spacecraft guidance, GUIDEACT, may drop from FINE LOCK to FINE LOCK/GYRO, or even to GYRO, which may result in a target rolling out of an aperture. Check the OMS header keywords to verify that the guiding mode was that which was requested, or for the archive user, to verify that there was no change in the requested guiding mode during the observation.

Until new flight software (version FSW 9.6) came online in September 1995, if the guide star acquisition failed, the guiding dropped to COARSE track. After September 1995, if the guide star acquisition failed, the tracking did not drop to COARSE track. Archival researchers may find older datasets that were obtained with COARSE track guiding.

The dominant and roll guide star keywords (GSD and GSR) in the OMS header can be checked to verify that two guide stars were used for guiding. Or in the case of an acquisition failure, to identify the suspect guide star. The following list of .cmh keywords is an example of two star guiding.

```
GSD_ID  = '0853601369         ' / Dominant Guide Star ID
GSD_RA  =           102.42595 / Dominant Guide Star RA (deg)
GSD_DEC =           -53.41362 / Dominant Guide Star DEC (deg)
GSD_MAG =              11.251 / Dominant Guide Star Magnitude
GSR_ID  = '0853602072         ' / Roll Guide Star ID
GSR_RA  =           102.10903 / Roll Guide Star RA (deg)
GSR_DEC =           -53.77683 / Roll Guide Star DEC (deg)
GSR_MAG =              12.426 / Roll Guide Star Magnitude
```

If you suspect that a target has rolled out of the aperture during an exposure, you can quickly check the counts in each group of the raw science data. As an example, the following IRAF commands can be used to determine the counts in each group.

```
% cl
> grlist z2o4040dt.d0h 1-24 > groups.lis
> imstat @groups.lis
```

Some observations can span several orbits. If during a multiple orbit observation the guide star reacquisition fails, the observation may be terminated with possible loss of observing time, or switch to other less desirable guiding modes. The GSACQ keyword in the .cmh header will state the time of the last successful guide star acquisition.

```
GSACQ   = '136:14:10:37.43   ' / Actual time of GS Acquisition Completion
```

Guide Star Acquisition Failure

The guide star acquisition at the start of the observation set could fail if the FGS fails to lock onto the guide star. The target may not be in the aperture, or for an extended target, only a piece of the target may be in the aperture. The jitter values will be increased because FINE LOCK was not be used. The following list of .cmh header keywords indicate that the guide star acquisition failed.

```
V3_RMS =                 19.3 / V3 Axis RMS (milli-arcsec)
V3_P2P  =               135.7 / V3 Axis peak to peak (milli-arcsec)
GSFAIL = '           DEGRADED' / Guide star acquisition failure!
```

The observation logs for all of the following observations in the observation set will have the "DEGRADED" guide star message. This is not a "Loss of Lock" situation, but is an actual failure to acquire the guide star in the desired guiding mode. For the example above, the guiding mode dropped from FINE LOCK to COARSE TRACK.

```
GUIDECMD= 'FINE LOCK          ' / Commanded Guiding mode
GUIDEACT= 'COARSE TRACK       ' / Actual Guiding mode at end of GS acquisition
```

If the observational dataset spans multiple orbits, the guide star will be re-acquired, but the guiding mode will not change from COARSE TRACK. In September 1995, the flight software was changed so that COARSE TRACK is no longer an option. The guiding mode drops from two guide star FINE LOCK to one guide star FINE LOCK , or to GYRO control.

Moving Targets and Spatial Scans

A type 51 slew is used to track moving targets (planets, satellites, asteroids, and comets). Observations are scheduled with FINE LOCK acquisition, i.e., with two or one guide stars. Usually, a guide star pair will stay within the pickle during the entire observation set, but if two guide stars are not available, a single guide star may be used, assuming the drift is small or the proposer says that the roll is not important for that particular observing program. An option during scheduling is to drop from FGS control to GYRO control when the guide stars move out of the FGS. Also, guide star handoffs (which are not a simple dropping of the guide stars to GYRO control) will affect the guiding and may be noticeable when the jitter ball is plotted.

The jitter statistics are accumulated at the start of the observation window. Moving targets and spatial scan motion will be seen in the jitter data and image. Therefore, the OMS header keywords V2_RMS and V3_RMS values (the root mean square of the jitter about the V2 and V3 axis) can be quite large for moving targets. Also, a special anomaly keyword (SLEWING) will be appended to the OMS header stating movement of the telescope during the observation. This is expected for observing moving targets. The following list of .cmh header keywords is an example of expected values while tracking a moving target.

```
                              / LINE OF SIGHT JITTER SUMMARY
V2_RMS  =                 3.2 / V2 Axis RMS (milli-arcsec)
V2_P2P  =                17.3 / V2 Axis peak to peak (milli-arcsec)
V3_RMS  =                14.3 / V3 Axis RMS (milli-arcsec)
V3_P2P  =                53.6 / V3 Axis peak to peak (milli-arcsec)
RA_AVG  =           244.01757 / Average RA (deg)
DEC_AVG =           -20.63654 / Average DEC (deg)
ROLL_AVG=           280.52591 / Average Roll (deg)
SLEWING = `                T' / Slewing occurred during this observation
```

High Jitter

The spacecraft may shake during an observation, even though the guiding mode is FINE LOCK. This may be due to a micro-meteorite hit, jitter at a day-night transition, or for other unknown reasons. The FGS is quite stable and will track a guide star even during substantial motion. The target may move about in an aperture, but the FGS will continue to track guide stars and reposition the target into the aperture. For most observations, the movement about the aperture during a spacecraft excursion will be quite small, but sometimes, especially for observations with the spectrographs, the aperture may move enough that the measured flux for the target will be less than a previous group. Check the OMS header keywords (V2_RMS, V3_RMS) for the root mean square of the jitter about the V2 and V3 axis. The following list of `.cmh` header keywords is an example of typical guiding rms values.

```
                              / LINE OF SIGHT JITTER SUMMARY
V2_RMS  =                 2.6 / V2 Axis RMS (milli-arcsec)
V2_P2P  =                23.8 / V2 Axis peak to peak (milli-arcsec)
V3_RMS  =                 2.5 / V3 Axis RMS (milli-arcsec)
V3_P2P  =                32.3 / V3 Axis peak to peak (milli-arcsec)
```

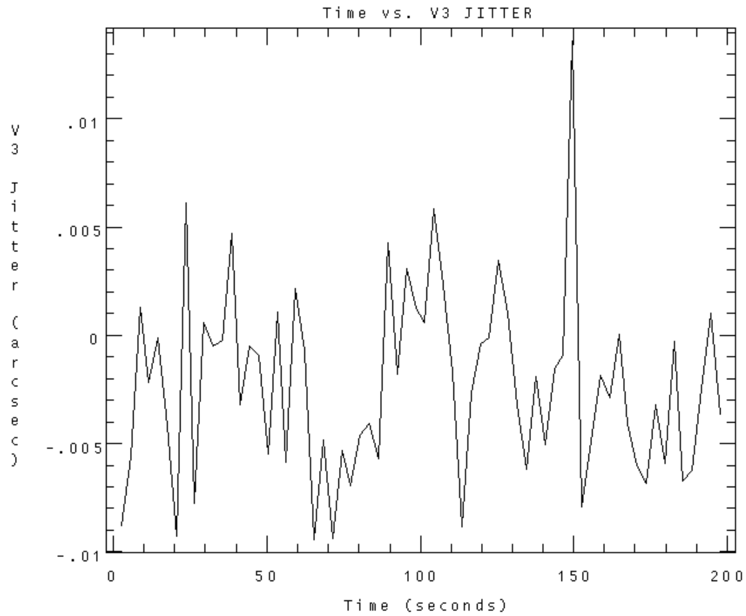
Recentering events occur when the spacecraft software decides that shaking is too severe to maintain lock. The FGS will release guide star control and within a few seconds reacquire the guide stars. It is assumed the guide stars are still within the FGS field of view. During the recentering time, INDEF will be written to the OMS table. Recentering events are tracked in the OMS header file.

Be careful when interpreting “Loss of Lock” and “Recentering” events that occur at the very beginning or at the end of the OMS window. The OMS window is larger than the observation window. These events might not affect the observation since the observation start time will occur after the guide stars are acquired (or re-acquired), and the observation stop time may occur before the “Loss of Lock” or “Recentering” event that occurred at the end of an OMS window.

The following IRAF command shows how to plot time vs. jitter along the direction of the V3 axis (see Figure 5.4):

```
cl> sgraph "u26m0801j.cmi seconds si_v2_avg"
```

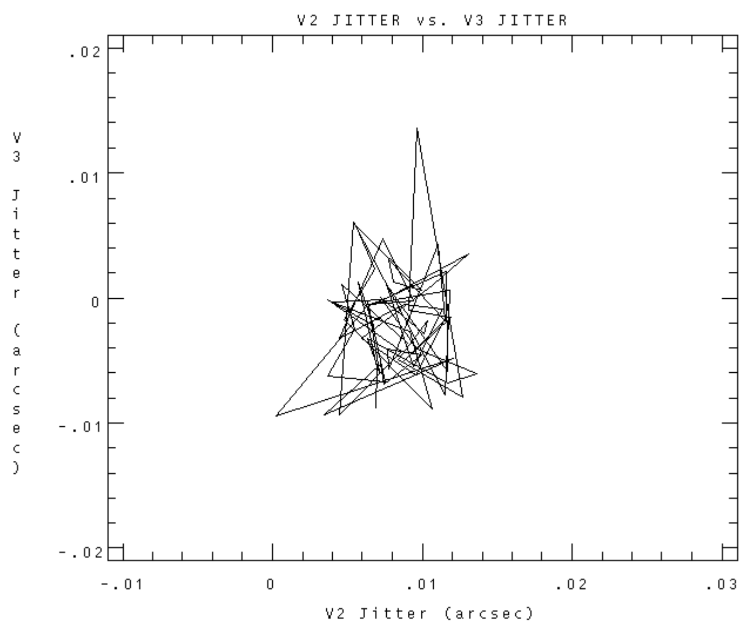
Figure 5.4: Plotting Jitter Along V3 Axis



To get an idea of pointing stability, plot jitter along the V2 axis vs. jitter along the V3 axis (see Figure 5.5):

```
st> sgraph "u26m0801j.cmi si_v2_avg si_v3_avg"
```

Figure 5.5: Plotting V2 vs. V3 Jitter



The **tstatistics** task can be used to find the mean value of the `si_v3_avg` column—the amount of jitter (in arcseconds) in the direction of the V3. This value could be used in software such as TinyTim to model jitter in a PSF. In this example, the mean jitter is ~3 mas, which is typical for post-servicing mission data:

Figure 5.6: Averaging a Column with tstatistics

```
tt> tstat u26m0801j.cmi si_v3_avg
# u26m0801j.cmi  si_v3_avg
# nrows            mean       stddev       median           min         max
   11   -0.003006443888  0.00362533  -7.17163E-4  -0.00929515  0.00470988
```

Understanding and interpreting the meaning of the table columns and header keywords is critical to understanding the observation logs. Please read the available documentation and contact the STScI Help Desk (`help@stsci.edu`) if you have any questions about the files. Documentation is available via the WWW at the following URL:
`http://ra.stsci.edu/documents/OL/OL_1.html`