Users' Guide to FTOOLS

Release 3.6

The FTOOLS Group

High Energy Astrophysics Science Archive Research Center
NASA Goddard Space Flight Center

October 21, 1996

1 Introduction

FTOOLS, a highly modular collection of over 200 utilities for processing and analyzing data in the FITS (Flexible Image Transport System) format, has been developed in support of the HEASARC (High Energy Astrophysics Science Archive Research Center) at NASA's Goddard Space Flight Center. Each utility performs a single simple task such as presentation of file contents, extraction of specific rows or columns, appending or merging tables, binning values in a column or selecting subsets of rows based on a boolean expression. Individual utilities can easily be chained together in scripts to achieve more complex operations such as the generation and displaying of spectra or light curves. The collection of utilities provides both generic processing and analysis utilities and utilities specific to high energy astrophysics data sets used for the ASCA, EINSTEIN, ROSAT, GRO, VELA5B and XTE missions. A core set of FTOOLS providing support for generic FITS data processing, FITS image analysis and timing analysis can easily be split out of the full software package for users not needing the high energy astrophysics mission utilities. The FTOOLS software package is designed to be both compatible with IRAF and completely stand-alone in a UNIX or VMS environment. The user interface is controlled by standard IRAF-style parameter files. The package is self documenting through the IRAF help facility and a stand-alone help task. Software is written in ANSI C, FORTRAN and Perl to provide portability across most computer systems. The data format dependencies between hardware platforms are isolated through the FITSIO library package.

This document describes the use of the FTOOLS software. The reader of this **Users' Guide to FTOOLS** will need to have some understanding of Unix or VMS login scripts to successfully use the Host version and a familiarity with IRAF to use the IRAF version.

Online HTML versions of this manual, tutorials, latest release information, the FAQ, and other goodies are available on the World Wide Web at:

`http://heasarc.gsfc.nasa.gov/docs/software/ftools/ftools_menu.html`

2 Usage

This section includes any information that individual users need to know to be able to use either the IRAF or Host versions of FTOOLS. It is assumed that FTOOLS have already been installed as described in the **Installation Guide to FTOOLS**.

2.1 IRAF Environments

There is very little that the user needs to do in order to use FTOOLS in the IRAF environment. If the user has not already done so, the standard IRAF user setup should be made. To do this the user should log into their account on a system where IRAF is installed. Then create a directory under the home directory in this account named iraf with the unix command:

```
>mkdir iraf
```

Next change directory to the newly created iraf directory with the command:

```
>cd iraf
```

Setup IRAF for this account by issuing the command:

```
>mkiraf
```

The user will be asked for a terminal type. Enter the terminal type associated with the terminal you will typically use to start up IRAF. This will cause a script file named login.cl to be created under the iraf directory, along with a new subdirectory named uparm to contain IRAF user parameter files. To start up IRAF the user simply enters the command:

```
>cl
```

from within the iraf directory. Once IRAF has started up a list of available packages are displayed. Among these is the ftools package. To load a package simply enter its name:

```
cl>ftools
```

This will then display a list of all the FTOOLS sub-packages. As of release 3.6 there are eleven sub-packages:

- ASCA
- CALTOOLS
- EINSTEIN
- FIMAGE
- FUTILS
- GRO
- HEASARC
- ROSAT
- TIME
- VELA5B
- XRONOS
- XTE

The ASCA package contains tasks useful for the ASCA mission, the CALTOOLS package contains tasks associated with calibration, the EINSTEIN package contains tasks useful for the Einstein mission, the FIMAGE package contains tasks useful for images, the FUTILS package contains general purpose FITS utility tools which apply to FITS data files from any specialty, the GRO package contains tasks designed for use with data from the Compton Observatory, the HEASARC package contains tools specific to high energy astrophysics data sets, the ROSAT package contains tasks useful for the ROSAT mission, the TIME package contains tasks useful for handling time, the VELA5B package contains tasks specifically designed for use with data from the VELA5B mission, the XRONOS package contains tasks for general timing analysis, and the XTE package contains tasks useful for the XTE mission. To load a package simply enter its name:

```
ft>futils
```

To get a simplified one line statement of the purpose of each futils task by entering the command:

```
fu>help
```

If more detailed help is needed for a particular task enter the help command followed by the name of the task:

```
fu>help fdump
```

The online help also provides examples to assist the IRAF novice.

Several of the FTOOLS are declared as external tasks within IRAF and will not be available for use in IRAF unless the HOST system version of the FTOOLS is installed and the users have their account properly setup to use the HOST version. These are `faintdfe`, `fplot`, `gqaplot`, `pconfig`, `pget`, `plist`, `pquery`, `pset`, `punlearn`, `sisclean`, `sisgbr`, `sqaplot`, and `brcaldb`. Similarly, there are a number of useful Perl scripts which are part of the HOST version of FTOOLS; these are also implemented as external or “foreign” tasks within IRAF.

2.2 HOST Environments

2.2.1 Unix

Using the Host version of FTOOLS will require several bits of information that should be provided by the system administrator that performed the Host installation. To use the Host version of FTOOLS the user must know the path to the location of the Host executable files and the parameter files. If FTOOLS was installed in the ftools account as described in the **Installation Guide to FTOOLS**, then this path is probably `~ftools/ftools/bin.host/`. Otherwise contact the system administrator for this information. Setting up the user's account to use the Host version of FTOOLS can be accomplished manually or with a script as discussed below. The script set up is described first.

2.2.1.1 `mkftools` Script

All of the necessary setup for using the FTOOLS can be conducted under the control of a script file provided with this release of FTOOLS. The required script is located in the `~ftools/ftools/bin.host` directory and is called `mkftools`. Check with the system administrator for the exact location of this script and the FTOOLS executables. This is all that the user needs to know before using this script. The easiest way to use the script is to copy it over to your user directory.

```
>cp ~ftools/ftools/bin.host/mkftools .
```

Then just execute the script by typing its name.

```
>mkftools
```

The script will instruct you on how to give it the proper arguments for your local setup. Type mkftools again with the appropriate argument for your site. The mkftools script will then step you through the process of building a script file called `.ftools` that will exist in your home directory and be executed at login. This script will also add the necessary line to .login to source the file at login time. Note that this script should only be run once and all previous FTOOLS setup should be removed from the user's login scripts to localize any potential problems.

2.2.1.2 OPTIONAL: Manual Setup Procedure

You can perform all of the script functions manually if you really want to. The user must specify the path to the executables with a command like:

```
>set path = $path:~ftools/ftools/bin.host
```

where the actual path should be used in the place of ~ftools/ftools/bin.host. This command should be added to the user's .login file to append the bin.host path to the previously specified path.

Since the release of FTOOLS 3.2, the language Perl has been standardized for FTOOLS scripts. The FTOOLS distribution now places Perl scripts in the `~ftools/ftools/bin.perl` directory. If your system has Perl installed and you would like use these scripts, which include the very popular `ascascreen` script (which now works on all supported UNIX systems), you'll want to prepend this directory to your path as with the `bin.host` directory:

```
>set path = ~ftools/ftools/bin.perl:$path
```

If the .login script file is altered it will need to be sourced with a command like:

```
>source .login
```

before the changes to these files will take effect. There are also four environment variables which must also be set up, LOCPFILES, SYSPFILES, PFILES and PFCLOBBER. But before establishing these make a directory under the user's home directory named something like `myftools` with the command:

```
>mkdir myftools
```

Set the current directory to the new myftools directory with the command:

```
>cd myftools
```

Now make two new directories under myftools. To do this give the command:

```
>mkdir pfiles
```

Next the user must set the environment variables. This can be done with the following commands assuming that the directory structure for the user's account follows the examples above:

```
>setenv LOCPFILES ~/myftools/pfiles
>setenv SYSPFILES ~ftools/ftools/bin.host
>setenv PFILES "$LOCPFILES;$SYSPFILES"
>setenv PFCLOBBER 1
```

You will also need to tell the perl scripts where to find the FTOOLS perl library routines. Prepend the path to the `bin.perl` directory to the PERLLIB environment variable. If the PERLLIB environment variable does not exist set it equal to the `bin.perl` path. If you have a PERL5LIB variable already set, use that instead as it overrides the PERLLIB variable.

```
>setenv PERLLIB = ~ftools/ftools/bin.perl:${PERLLIB}
```

if you already have a PERLLIB path, or

```
>setenv PERLLIB = ~ftools/ftools/bin.perl
```

if you don't, or

```
>setenv PERL5LIB = ~ftools/ftools/bin.perl:${PERL5LIB}
```

if you have a PERL5LIB path set.

The user should add these commands to their .login script file so the variables will be set up during each login. Remember that the file which is changed must be sourced afterwards to set the environment variables or the user can logout and then log back in. Once this procedure has been completed the user can go to any directory and use the Host Version of the FTOOLS tasks. See Appendix A for a list of FTOOLS tasks in this release.

2.2.2 VMS

To use the Host version of FTOOLS, a VMS command file has been provided to set up all of the logical definitions that a user will need. Check with your system administrator to see if a version of this command file is already available with your system's specific information. It is preferred to place a common copy of this file in a place for all users to access. If not, and assuming that the FTOOLS were installed as described in the **Installation Guide to FTOOLS**, copy the command file to your local directory.

```
$copy [ftools.ftools.scripts.vms]init_ftools.com *.*
```

Next, edit this file to change the specification of the location of FTOOLS package and PERL, if available. The command file checks whether the user is on a VAX or an ALPHA, and whether a release or development versions is requested. The appropriate version of FTOOLS is then selected. Change the location(s) of the flavor of FTOOLS installed on your system as indicated in the file. Note that you must use the true disk name in this command. Specifying a concealed logical in a concealed logical has unpredictable results in VMS. Also be sure to include the trailing . in the directory specification. The file can now be run using:

```
$@init_ftools
```

This file must be executed before the user can run the FTOOLS, so it is recommended that the command be added to the user's login.com. In addition to setting up all of the logical values needed, the file will create a directory, `[youraccount.ftools.pfiles]` to hold the users parameter files. A command is also defined to "fix" the attributes of FITS files which are incorrectly set by FTP, VMSTAR and UNCOMPRESS. FFFILE (Fix Fits FILE) will change the record length of the FITS file so that it can be read more efficiently.

If you prefer to set all of the necessary logicals by hand, the following commands will need to be issued before running FTOOLS:

Define ftools to be a logical disk:

```
$assign/trans=conceal casper$dka100:[ftools.ftools.]  ftools
```

Define all of the FTOOLS to be foreign commands:

```
$@ftools:[script.vms]learn ftools:[bin_host]
```

Define logicals needed for plotting routines:

```
$assign ftools:[host]libpgplot.exe libpgplot
```

```
$assign ftools:[host]grfont.dat PGPLOT_FONT
```

Tell the ftools about the location of your local copy of the parameter files, and the location of the default parameter files:

```
$assign "mydisk:[myaccount.parfiledir];ftools:[bin_host]" pfiles
```

And finally, indicate that you would like the FTOOLS to remember the previous values you entered for parameters:

```
$assign 1 pfclobber
```

If you are using the perl scripts, you should first tell VMS where to find perl as described in the VMS perl installation documents (whoever installed perl on your system should have done this in a way that will work for you). To test that perl is working from your account type:

```
$perl -e "print 'Hello World';"
```

Assuming this works, you must define all of the perl scripts to be foreign commands:

```
$@ftools:[scripts.vms]knowperl ftools:[bin_perl]
```

You will also need to tell the perl scripts where to find the FTOOLS perl library routines. Prepend `/ftools/bin_perl` (the UNIX style pathnames are required) to any existing PERLLIB logical environment variable. If PERLLIB does not exist, set it equal to the `/ftools/bin_perl`. If you have a PERL5LIB variable already set, use that instead as it overrides the PERLLIB variable.

`$define PERLLIB= "/ftools/bin_perl:" +` *whatever PERLLIB was before*

if you already have a PERLLIB variable, or

```
$define PERLLIB "/ftools/bin_perl"
```

if you don't, or

`$define PERL5LIB= "/ftools/bin_perl:" +` *whatever PERL5LIB was before*

if you have a PERL5LIB variable set.

Once this procedure has been completed, the user can go to any directory with read and write privileges and use the Host Version of the FTOOLS tasks. See Appendix A for a list of FTOOLS tasks in this release.

3 FHELP

FHELP is a Host task which provides the user with IRAF like Help pages for FTOOLS. Users of the stand-alone Host version of this task automatically have access to a collection of ASCII formatted help files that are distributed in the `~ftools/ftools/help/` directory (for VMS, `ftools:[help]`) of FTOOLS. FHELP is a perl script unless perl isn't installed at your site, in which case, FHELP is a compiled FORTRAN program. Type 'fhelp fhelp' to get going. FHELP presents the task help a page at a time. If you are using the Perl version of fhelp, the output will be sent through your system's "more" utility ("type/p" on VMS). If you are using the compiled fhelp, pressing `y<return>` pages the help file, pressing `n<return>` quits the display of the help file.

4 FTOOLS that Plot

FPLOT, GQAPLOT, SQAPLOT and SISCLEAN are tasks within the FTOOLS package for plotting datasets taken from FITS tables. Most of the tasks in the Xronos subpackage also plot. All these tasks are based on the QDP/PLT plotting program by Allyn Tennant of NASA's Marshall Space Flight Center. QDP/PLT is itself based on PGPLOT by T.J. Pearson of California Institute of Technology. Because of the multiple layers of software dependencies found under these tasks, not all sites may be able to install them. All of the necessary source code is included, so an industrious system administrator may be able to complete the installation. (If this is the case, please send the necessary changes back to the FTOOLS group.) IRAF users should not despair, all can be linked with the rest of the FTOOLS package under IRAF as an external system task, complete with an IRAF like parameter interface and help pages. In IRAF, the Xronos tasks are built as native IRAF tasks; however, their plotting capability is currently disabled. In addition, SQAPLOT (and SISRMG) are not available on VMS or the ALPHAs at this time.

To use FPLOT, GQAPLOT, SQAPLOT and SISCLEAN follow the directions in section 2.2 of this document. The FTOOLS script is configured to recognize FPLOT, GQAPLOT, SQAPLOT and SISCLEAN when they exist. In order to have these tasks label the plots produced, the environment variable `PGPLOT_FONT` must be defined. This variable must point to the `grfont.dat` file. It is set in the mkftools and init_ftools scripts. If you are not running the appropriate script, it is recommended that the following line be added to your `.login` or `.cshrc` file:

```
>setenv PGPLOT_FONT ~ ftools/ftools/host/grfont.dat
```

or for VMS:

```
$assign ftools:[host]grfont.dat pgplot_font
```

Serious users of FTOOLS that plot should consult the **QDP/PLT User's Guide** by Allyn Tennant, NASA Technical Memorandum 4301. A copy of this manual can be found on the FTOOLS web site in the "other documents" area:

`http://heasarc.gsfc.nasa.gov/docs/software/ftools/others.html`

5 The Calibration Database

A number of FTOOLS tasks require input calibration datasets. Many such tasks include software which (optionally) enables the name and location of the appropriate calibration files(s) to be supplied to the task (without user interaction) via a local Calibration Database (Caldb). Specific instructions as to how users can use their local Caldb are contained within the help pages for the task. If a local Caldb is not installed, again the help pages for the task instructs users as to what they must do. Obviously this usually consists of the user having to determine the name and location of the necessary calibration files for themselves.

For a variety of reasons, a local Caldb is **NOT** automatically set up by the installation of FTOOLS. However setting up your own local Caldb it is relatively trivial in most cases, and full instructions

are provided in the OGIP memo *How to Install a Calibration Database* (CAL/GEN/94-004) which is available via anonymous FTP to
`ftp://heasarc.gsfc.nasa.gov/caldb/docs/manuals/installation/install_guide.ps`
and on the WWW via the Caldb Home Page:
`http://heasarc.gsfc.nasa.gov/docs/heasarc/caldb/caldb_intro.html`

Since a local Caldb can become quite large, we recommend that a site maintain only a single database on a central file system and that this local Caldb be made available to multiple users. Before installing a Caldb, it's certainly worth checking one is not already available to you locally.

If/once a local Caldb is available, then a number of environment variables need to be set in order that FTOOLS tasks are able to locate the necessary database files *etc*. Assuming you have a standard set up, this can be done by sourcing a single file named `caldbinit.unix` for unix operating systems and `caldbinit.vms` for VMS. To find out where the appropriate file resides at your site, you should contact the person responsible for maintaining your local Caldb. (If you know where the top-level of your local Caldb resides, and have a standard set up, then the file is probably located in `caldb/software/tools/` directory for unix systems and `CALDB:[SOFTWARE.TOOLS]` for VMS systems.) Since the environment variables are required by all processes which need to access your local Caldb, we recommend you add a line to your login file:
`source {path}/caldbinit.unix` in your `.login` for unix systems, or
`$ @{path}CALDBINIT.VMS}` in your `LOGIN.COM` for vms systems,
where `{path}` is the path to the appropriate `caldbinit` file. Full instructions, further information and help are provided in the OGIP memo *The Calibration Database Users' Guide* (CAL/GEN/94-002) which is available via anonymous FTP to
`ftp://heasarc.gsfc.nasa.gov/caldb/docs/manuals/users/users_guide.ps`
and on the WWW via the Caldb Home Page:
`http://heasarc.gsfc.nasa.gov/docs/heasarc/caldb/caldb_intro.html`

Any questions/comments regarding the installation and operation of a local Caldb should be addressed to

`caldbhelp@athena.gsfc.nasa.gov`

6 A Brief FTOOLS Tutorial (Host Version)

This section provides an introduction to using the Host version of FTOOLS on a UNIX machine. Except for the difference in path syntax, the VMS version of FTOOLS is nearly identical to UNIX. Users of the IRAF version of FTOOLS should have little trouble getting started if they are already familiar with IRAF since FTOOLS behaves just like any other IRAF package. Novice IRAF users should contact their IRAF system manager for documentation on how to use IRAF.

6.1 Getting Help

General help information about FTOOLS as well as detailed help on each individual task can be obtained by the fhelp task. Type,

```
> fhelp ftools
```

to get a brief description of the FTOOLS package. To get detailed help on any particular FTOOLS task type,

```
> fhelp <taskname>
```

where <taskname> is the name of the task you are interested in. The help files give a description of what each task does and a description of all the required and optional (hidden) parameters. It is important to realize that when running an FTOOLS task you will only be required to enter values for the most important parameters, such as the name of the input FITS file. But most of the tasks also have other 'hidden' parameters which have a predefined value unless you explicitly reset them to a new value. These hidden parameters can significantly alter the operation of the task, so you need to be aware of what hidden parameters are available for a given task and what their default values are. One way to determine this is to read the help file for the task with the fhelp command, but a quicker way to simply see all the task parameters and their current values is with the plist command (analogous to the lpar command in IRAF). For example:

```
> plist fkeyprint

  infile =                 Name of FITS file and [ext#]
  keynam =                Enter the keyname (8 characters or less)
(outfile = STDOUT)        Name of optional output file
  (exact = no)            Exact string or not
(clobber = no)            Overwrite existing output file?
   (mode = ql)
```

shows that the fkeyprint task has 4 hidden parameters (by convention, the names of hidden parameters are enclosed in parentheses) to control the name of an optional output file where the keyword value should be written, whether to do string matching exactly or to match only as many characters as given, whether to overwrite an existing output file, and the mode parameter which determines how non-hidden parameters should be handled. By changing mode to 'h' you can "hide" all of your parameters, for instance. This can be dangerous and is not recommended for casual users.
A prompt for a parameter will generally display the default value for that parameter based on what was entered the last time the tool was run or was set using pset. Hitting "return" will select this value. Some FTOOLS tasks (particularly within the Xronos sub-package) will also calculate a default value for some of their parameters based on other input parameters. This default can be selected by typing "INDEF"

6.2 Running a task

The simplest way to run a task is to enter its name on the input command line followed by a <CR>, as in,

```
> fdump
```

The task will then prompt you for the value of all the required parameters. It will first display the current value of each parameter; if that value is OK then you can simple hit <CR> to accept it, otherwise enter the new value and hit <CR>. Note that you will not be prompted for any hidden parameters and the task will simply assume the default values (which can be listed with the plist command as shown above).

If you know the value of any or all of the required task parameters, you may alternatively enter their value directly on the command line in the same order as listed by plist. The task will then only prompt you for the values of any remaining non-hidden parameters which were not entered on the command line. The file `crabk.fits` can be found in the `testdata` directory. For example,

```
> fdump crabk.fits STDOUT RATE 1-20
```

will list the first 20 rows of the 'RATE' column in the FITS file crabk.fits onto the standard output device (STDOUT). As before, the task will assume the default value for any hidden parameters.

You may override the default value of a hidden parameter in one of 3 ways: by specifying the new value on the command line, by using the pset command or by editing the par file. In the first method, the specification of the hidden parameters must follow any required parameters that are also given on the command line, as in,

```
> fdump crabk.fits STDOUT prhead=yes prdata=no
```

This supplies values for the first 2 parameters, just like in the previous example, and then specifies that the 'prhead' parameter should have the value 'yes' and the 'prdata' parameter should have the value 'no'.

The other way to specify the value of any parameter, including the hidden parameters, is through the pset command.

> `pset fdump`

will prompt you in sequence for the value of every parameter in the fdump task. The value of an individual parameter may also be defined by specifying the name and value of the parameter on the command line.

> `pset fdump skip=5`

This example sets the skip parameter in the fdump task equal to 5. Note that the pset command does not execute the task, but instead merely sets the default parameter values for the all subsequent runs of the task. To reset all parameters to their default value, use

> `punlearn fdump`

6.3 Example FTOOLS Session

The following example will display the contents of a sample FITS file, then will extract some of the data into a new FITS file.

To run this example, first copy the sample FITS data file that is distributed with FTOOLS into a convenient work directory:

> `cp ~ftools/ftools/testdata/crabk.fits .`

(If this does not work, contact your system administrator for the path to the FTOOLS installation at your site).

In order to see what information is contained in the FITS file, the first step is to determine the basic structure of the file with the fstruct task:

> `fstruct crabk.fits`

```
 No. Type     EXTNAME      BITPIX Dimensions(columns)      PCOUNT  GCOUNT

   0  PRIMARY                -32     0                          0     1
   1  BINTABLE SPECTRUM        8     12(4) 128                  0     1
```

This shows that the FITS file crabk.fits consists of a primary array with zero data elements (i.e., the primary array consists only of header records, with no data array) followed by a single binary table extension called 'SPECTRUM' which has 4 columns of data (with a total of 12 bytes of information in each row) and 128 rows.

Next, the flcol task (List COLumns task) can be used to get more information about the names and datatypes of all the columns in a particular table:

> `flcol crabk.fits`

```
___Column_Names_________Formats__________Units___
CHANNEL                    I
RATE                       E           COUNTS/S
STAT_ERR_RATE              E           COUNTS/S
DQF                        I
```

Finally, the fdump task may be used to view the actual header and data contents of the FITS file:

> `fdump crabk.fits prhead=yes prdata=no`

This will list all headers in the FITS file (hit <CR> to accept the default values of the other required parameters). Note that the convention used in many the FTOOLS tasks is to operate on the first extension in the FITS file by default unless another extension is explicitly specified in square brackets appended to the input file name or following a + sign. For tasks explicitly dealing with images the primary array is the default. For instance, to print the header of the initial primary array, enter

> `fdump 'crabk.fits[0]' or fdump crabk.fits+0`

Most tasks operate on a single extension at a time, but some tasks, including fdump, can operate on all the extensions in a file in a single run, by specifying the extension number as [*]. Thus to display all the header records in the whole FITS file, type,

> `fdump 'crabk.fits[*]' prdata=no`

Now to display the first 20 rows of data in the table type,

> `fdump crabk.fits prdata=yes prhead=no rows=1-20`

As before, just hit <CR> to accept the default values of the other parameters. This should display the first 20 rows of data for all 4 columns of the table. Note that the first 14 rows of the data are undefined as indicated by the INDEF symbol.

To illustrate the use of a couple more FTOOLS tasks, we can modify the contents of this sample FITS file, to produce a new file. Suppose we want to create a new FITS table that only contains the rows which have defined data values (i.e., excluding the first 14 rows in the crabk.fits table. This can be done with the fselect task by selecting only those rows which have a value of 1 in the 'DQF' column:

> `fselect crabk.fits new.fits DQF==1`

This then creates a new output FITS file with a binary table extension containing only the rows from the input table which have DQF equal 1. This may be verified by typing:

> `fdump new.fits+1 rows=1-`

This concludes the demonstration of FTOOLS, which has hopefully illustrated the basic techniques that are needed to explore the capabilities of the other FTOOLS tasks.

A more descriptive discussion of the FTOOLS usage can be found in the article "FTOOLS Tutorial" in *Legacy* - The Journal of the HEASARC, 4, 5, 1994.

7 Using XSELECT

XSELECT is a high level command interface to the FTOOLS. It uses commands that are relevant to high energy astrophysics data analysis to chain together sequences of FTOOLS in a manner transparent to the user. This release includes a manual for XSELECT. It should be available wherever you found this manual, or go to the WWW site mentioned above.

In order for a user to use XSELECT, the Host version of FTOOLS must be installed on the system and the user must set up their account to properly use the Host version of the FTOOLS as discussed in the **Host Environments** section of this document. In particular, it is highly recommended that users set up their accounts to use the FTOOLS by running the `mkftools` script found in the `scripts/install/` subdirectory of the FTOOLS directory path. The init_ftools.com script will set up the necessary things for VMS users.

Using the full compliment of commands available to XSELECT users also requires that the image display program `saoimage`, available from SAO, be installed. Check with your system administrator to verify the existence of this program at your site. Binary versions of this program are available via anonymous FTP from the software archive at the HEASARC.

8 PERL Scripts

Since the version 3.2 release, FTOOLS has included a suite of PERL scripts. PERL has been adopted as the standard scripting language for use in the FTOOLS. To maximize portability, PERL 5.0 or above is recommended (and required for VMS, see Installation Guide). As described in the Installation Guide, unpatched versions of Perl5.001 may not work properly. If you experience problems with Perl FTOOLS, you may need to upgrade to a more recent version of Perl (version 5.002 as of this writing). The FTOOLS distribution includes a `bin.perl/` directory where these scripts reside after the FTOOLS build is completed. If your system doesn't have PERL the scripts will call a dummy program called "noperl" which prints a blurb stating the scripts can't run. UNIX users who obtain perl at later time can change this with the find_perl.sh script in `bin.perl/`. Simply set the environment variable PERL to the installed perl (with the full pathname) and then execute find_perl.sh, with a wild card or specific script name. Assuming you're in `bin.perl/` and using the csh, the commands would be:

```
>setenv PERL /some_full_pathname/perl
>find_perl.sh *
```

This will fix all of the perl scripts in `bin.perl/` to use the new perl. VMS users should consult their system manager for assistance after a perl installation.

To use the PERL scripts found in the `bin.perl/` directory you must add the directory (full path) to your PATH environment variable. This is most easily accomplished using the **`mkftools`** script which is located in the `bin.host/` directory of FTOOLS. The init_ftools.com script will perform the analogous things for VMS users.

9 Appendix

A FTOOLS Task Summary

The current release of FTOOLS has the following tasks and Perl scripts implemented within the twelve sub-packages. Additional help for each task can be found through the IRAF help facility or with the Host FHELP task once the FTOOLS package has been loaded.

ASCA specific FITS tasks:

- add_sis_rmf - [P] Reports percentage of counts in evtfile in each chip
- ascaarf - Generates an ASCA ARF file for an input PHA file
- ascaeffmap - Generates an ASCA EFF detector image file for an input PHA file
- ascaexpo - Generate an exposure map for an ASCA observation.
- ascalin - Produce the ASCA calibrated science file data products
- ascascreen - [P] Cleans raw ASCA science files
- ascatape - Read FITS files off NASA/GSFC/ADF supplied ASCA data distribution tapes.
- aspect - Computes the Euler angles of the best aspect point
- attitude - Compute summary information for an ASCA attitude file
- cleansis - Detect anomalous ASCA SIS pixel values and output the cleaned events file with a 'hot pixel' extension appended
- dfe2mkf - Insert output information of FAINTDFE into existing MKF file

- faint - Convert ASCA faint mode data to a bright mode format
- faintdfe - Calculate the dark frame error for SIS FAINT mode data ***
- fasttime - Corrects SIS FAST mode data times
- fastwmap - Add a WMAP to a FAST mode spectral file.
- ghkcurve - Extract light curve from ASCA GIS housekeeping file.
- ghkdump - Display GIS housekeeping parameters of the GIS HK file
- gisrti - Compute Rise Time Invariant value in GIS PH mode science file
- gqaplot - ASCA GIS quick analysis ftool ***
- ldeadtime - Carry out deadtime correction for FITS light curves.
- mkdtime - Calculates the deadtime fractions for the GIS
- modalid - Print the space craft modal configuration
- sisclean - Plot frequency histogram of an SIS image file and apply an intensity threshold cut ***
- sisgbr - [P] QDP fit branching ratios derived from FITS data
- sishist - Frequency histogram of an image created from an SIS event file
- sispi - Fills PI column of ASCA SIS event files.
- sisrmg - Generate SIS instrument response matrix
- sqaplot - ASCA SIS quick analysis ftool ***
- timeconv - Convert from satellite to barycentric (use with caution on SUN)

Calibration Database (Caldb) specific FITS tasks:

- addarf - Adds two or more Ancillary Response Files (ARFs)
- addrmf - Adds together two or more RMFs
- brcaldb - Browse the Caldb data holdings
- calcrpsf - Calculates a radial point spread function
- caldbinfo - Checks whether a local CALDB is correctly installed for a user
- caldbflag - switches flag status of calibration files in a CIF
- chkcif - Check entries in a Calibration Index File
- crcif - Creates an empty Calibration Index File
- cmppha - Converts a TYPE II to a TYPE I OGIP pha file
- col2img - Converts a collimator response dataset to an image
- dmprmf - Displays OGIP standard Response FITS file
- gcorpha - remaps PHA channels from SPECTRUM extension.
- gcorrmf - Remaps a detector redistribution matrix in channel-space

- lstgood - List "good" files found in a Calibration Index File
- marfrmf - Multiplies an ancillary response file by a redist matrix
- mkcaldb - Create a Calibration Database
- mkcaldir - Create the Caldb directory structure
- mkcalinit - Create a caldbinit file
- quzcif - Interrogates Caldb for location of a dataset
- rbnrpsf - Rebins a radial point spread function
- rpsf2eef - Converts a radial PSF (RPSF) dataset to a radial encircled energy function (REEF) dataset.
- rsp2rmf - Converts SF format response matrices to OGIP FITS format
- rpsfqdp - Converts OGIP format radial profile file to ASCII QDP
- rbnrmf - Physically compresses a RMF file
- st2rpsf - Reads STWFITS file and writes an OGIP RPSF file
- stcal - Stores one or more calibration files in the Caldb
- stw2pha - Converts stw PHA file to OGIP FITS format
- udcif - Adds a calibration file to a CIF

Einstein specific FITS tasks:

- sssarf - creates an Ancillary Response File (ARF) for Einstein SSS PHA data
- sssbck - builds the EINSTEIN SSS background PHA files for a given source obsn

Examples Package:

- cdummyftool - "Hello World" C ftool example/template
- fdummyftool - "Hello World" Fortran ftool example/template

FITS Image tasks:

- f2dhisto - Make a 2-D histogram from 2 columns in a table
- fadapt - Convolve a FITS image with an adaptive top hat filter
- farith - Perform arithmetic on 2 FITS images
- fboxcar - Convolve a FITS image with a rectangular boxcar filter
- fcarith - Perform arithmetic on FITS image with a constant
- fgauss - Convolve a FITS image with an elliptical Gaussian function
- fim2lst - Convert a 2D image to a pixel list (inverse of f2dhisto)
- fimconcat - Take 2 images and place them side by side as single image
- fimgbin - Rebin a FITS image using a rectangular bin size

- fimgdmp - Dump contents of a FITS image to an ASCII file
- florentz - Convolve a FITS image with an elliptical Lorentzian function
- fmaskfilt - Filter an event list based on an input mask image
- fmrgmsk - Merge 2 or more spatial masks
- fregcon - SAOimage region file converter from one instrument/detector to another
- fsaoi - Translate an SAOImage region file to an input file for fselect
- fvec2img - Create an image from a column of vectors

FUTILS - General Purpose FITS tasks:

- faddcol - Copies columns from one FITS table extension to another
- fappend - Append a FITS extension onto another FITS file
- fapropos - [P] Does keyword searches for FTOOLS
- fcalc - Calculates values for a column using an arithmetic expression
- fcatdiff - Compares columns of a fits file and reports row differences
- fchecksum - Verify or update the checksum keywords in a FITS file
- fcolpar - Returns column number for an input filename and column name
- fcreate - Create a FITS table from ASCII input files
- fcurve - Make a light curve histogram from a column in a table
- fdump - Dump contents of a FITS table to an ASCII file
- fextract - Copy a FITS extension from a file into a new file
- ffilecat - Copies keyword values from a list of FITS file to FITS Table
- fhelp - Display FTOOLS help information
- fhisto - Make a histogram of a column in a table
- fimgcreate - Create a FITS primary array image from an ASCII template file
- findex - Create an index file for a FITS table column
- finterp - Interpolate a column from one FITS table extension into another
- fkeypar - Copy a FITS header keyword to a parameter
- fkeyprint - Display keyword(s) in FITS header(s)
- fkeytab - Copy a FITS header keyword to a FITS table element
- flcol - List column information in a FITS table extension
- flist - List contents of a FITS table to an ASCII file
- flookup - Filter a FITS file from selection based on a lookup table
- fmemsort - Fast memory sort of a FITS table
- fmerge - Merge rows from several FITS tables into one FITS table

- fmodcomm - Modify commentary keywords in a FITS file header
- fmodhead - Modify the header keywords in a FITS file
- fparkey - Copy a parameter value to a FITS header keyword
- fparstamp - [P] Copy contents of a .par file into a FITS file header
- fpartab - Copy a parameter value to a FITS table element
- fplot - Plots columns from a FITS file using QDP/PLT Plot Package ***
- fselect - Create a new table from selected rows of a table
- fsort - Sort a FITS table in place
- fstatistic - Calculate mean, standard deviation, min, and max for a column
- fstruct - List a description of the structure of a FITS file
- ftabcopy - Copy specified columns of a FITS table to a new table
- ftabkey - Copy a FITS table element to a FITS header keyword
- ftabpar - Copy a FITS table element to a parameter value
- fv - An interactive GUI FITS file browser and display tool
- fverify - Verify that a FITS file conforms to the FITS Standard
- fversion - [P] Report FTOOLS version number and release date

GRO specific FITS tasks:

- fbbft2pha - Converts BATSE BFITS file to spectral PHA file
- fbdrm2rmf - Converts BATSE DRM file to RMF format
- fesdb2rdf - Reads i/p QVP and EXP (optional) data files and writes an o/p FITS data file in OGIP standard format
- fevpbtoa - Converts Fits formatted Comptel EVP data to an ASCII file

HEASARC specific FITS tasks:

- addspec - adds PHA spectra, bkgd files and combines response matrices
- ascii2pha - Creates an OGIP FITS PHA file from an ascii dataset
- bincurve - Bins vectors and outputs as a light curve
- binspec - Bin spectra in vector columns
- bkg - Has been replaced by the tool lcmath in the Xronos package
- ecd2pha - Converts Einstein HEASARC CD-ROM PHA to OGIP PHA file
- extrpsf - ExtractS radial PSF (RPSF) dataset from an event file.
- fadmap - Produce maps for SAS2 or COSB data
- fixregion - image region file converter from the image file it was extracted, to another instrument/detector system.

- grppha - Manipulates OGIP standard PHA FITS file
- hkexpand - Expand a compressed format housekeeping (HK) data file
- hkscale - Scales a FITS housekeeping data file into physical values
- hkunexpand - Compress an expanded format housekeeping (HK) data file
- mathpha - Performs mathematical operations on PHA files
- rbnpha - Compresses a PHA dataset
- sf2pha - Converts SF format PHA file to OGIP FITS format
- sky2xy - Transform from sky to pixel coordinates using WCS keywords
- xy2sky - Transform from pixel to sky coordinates using WCS keywords

ROSAT specific FITS tasks:

- abc - Apply Barycentric Corrections to ROSAT event data.
- bct - Produce a Barycenter Correction Table from ROSAT orbit data
- calcbgdcor - calculates the background spectra for a given source region with a given background region, away from source region.
- castpart - Cast model particle background counts in sky coordinates
- extpha - [P] Extracts source spectrum files from PSPC _src files
- fixrev0pha - Changes the CDELT1 and CDELT2 keywords in a ROSAT PHA file
- gevt2rdf - Converts a German _events dataset to RDF _bas format
- hiback - [P] Creates _qsrc file from _src file
- hriaspcor - Use Morse's method to correct aspect errors
- hriexpmap - Creates exposure map for given ROSAT HRI observational dataset
- hrifilt - Creates a ROSAT HRI makefilter file
- hrirpsf - Calculates theoretical HRI PSF
- img2us - Converts from German Image file format to US RFITS format
- mkqsrc - Adds extra fields for quality screening to ROSAT source file
- mk_allevt - [P] Combines standard and rejected events tables of an RDF file
- pcarf - Generates a ROSAT PSPC ARF for an input PHA file
- pcbgdcor - background subtraction [P]
- pcecor - Corrects PI channel in Events file
- pcexpmap - Creates exposure map for ROSAT PSPC observation
- pcfilt - Creates a ROSAT PSPC makefilter file
- pcparpha - Generates a background PHA spectrum for ROSAT PSPC
- pcpicor - [P] Corrects PI channel in Events file

- pcsasscor - Corrects PI channel in PSPC events file
- pcrpsf - Calculates a theoretical radial PSF for the ROSAT PSPC
- pctcor - Corrects PI channel in Events file
- rosbary - [P] Perform barycentric correction for ROSAT. (Runs abc and bct)
- roscc2utc - given the space craft clock time, it calculates the utc time in JD, MJD, yy:mm:dd:hr:min:sec
- src2pha - Creates a PHA file from specified extensions in an RDF src file

TIME specific FITS tasks:

- burst - Remove bursts of events from time ordered event list
- cktime - Checks that a time column is time ordered
- deadtime - Update the exposure of PHA file by deadtime based on GTIs
- fltime - Filter an event list within given time intervals (GTIs)
- maketime - Calculate time intervals (GTIs) from housekeeping (HK) data
- mgtime - Merge 2 or more time interval (GTI) files
- sec2time - Convert time offset to absolute time
- time2sec - Convert absolute time to a time offset

VELA5B specific FITS tasks:

- fvelalc - Creates a FITS light curve for an isolated Vela 5B source
- fvelmap - Creates a set of FITS source maps of a rectangular region
- fvelgallc - Creates multiple light curves for sources in confused fields
- velabox - Lists the Vela 5B raw data files needed for FVELALC
- velamap - Lists the Vela 5B raw data files needed for FVELMAP

XRONOS General Purpose FITS timing tasks:

- ascii2flc - [P] creates a xronos input FITS file from an ASCII file
- autocor - calculates auto correlation for a time series
- crosscor - calculates cross correlation of two time series
- earth2sun - change times to solar system barycenter
- efold - creates epoch folded lightcurve, hardness and color-color plots
- efsearch - finds best chi-squared period by folding over range of periods
- flc2ascii - [P] Dump xronos input or output in ASCII file readable by QDP
- lcmath - Subtracts or adds two binned lightcurves.
- lcstats - calculates statistical variables a time series
- lcurve - creates lightcurves, hardness and color-color plots

- listdata - list data file on the screen
- powspec - creates power spectrum density
- rbf2fits - Converts the "EXOSAT rate buffer" format in FITS.
- timeskew - calculates time skewness function for a time series
- xronwin - [P] creates a window file suitable for xronos tasks

XTE specific FITS tasks:

- addshots - Adds a shot noise light curve to an input light curve.
- addsine - Adds a sine curve to an input light curve.
- asmappend - Sorts and appends ASM files to a MASTERFILE.
- asmsrcupdate - [P] Runs ASMAPPEND on a collection of XTE ASM master files and their respective incoming files.
- bemerge - Merge two binned event spectral (.pha) files: type I or II.
- chantrans - [P] Translate relative channel range to an absolute channel range by reading XTE produced .pha files.
- edsgcor - Converts the PCA/EDS Gain and Offset file from ASCII to FITS.
- fakelc - Creates a fake light curve having a constant mean intensity.
- fcollect - First half of XTEFILT; collects housekeeping, attitude and orbit data from XFF database.
- fxbary - Apply barycentric correction light-curve (.lc) FITS file.
- grosstimefilt - [P] Filter out all times that fall outside of a specified range and modify appropriate keywords in a XTE SA or SE file.
- make_se - [P] Create XTE/SE (Event) data files from a list of input files which may contain Good_Xenon and/or Transparent Mode data files.
- pca_bs - [P] Perform rough PCA lightcurve background subtraction.
- pcabackest - Creates and estimated PCA background spectrum from model.
- pcaclrsp - Creates a collimator corrected light curve from PCA .lc file.
- pcagainset - Insert PCA/EDS gain and offset values into a PCA .pha file.
- pcarmf - Creates a .rmf file for a specified anode chain in a PCU.
- pcarsp - [P] Runs the tools necessary to construct response matrices for XTE PCA spectral data.
- perdgrm - Computes a periodgram for unevenly sampled light curves.
- rddescr - Parses a channel descriptor from an XTE .pha file to ASCII.
- runpcabackest - [P] Runs PCABACKEST upon a series of Std Mode 2 XTE data files.
- saexpd - Reads science array vectors and outputs generic FITS files.
- saextrct - Creates lightcurve and/or spectrum from science array vectors.

- sebitmask - Creates bitmask for filtering SE data with FSELECT.
- seexpd - Reads science events data and outputs generic FITS files.
- seextrct - Creates lightcurve and/or spectrum from science events data.
- sachip - [P] Runs SAEXTRCT and FPLOT once from command line.
- saplot - [P] Cyclically queries the user for additional SA files and then runs SAEXTRCT and FPLOT, for SOF.
- sefilter - [P] Helps the user generate an expression that is fed into SEBITMASK and examines the result to determine if the file can be directly input into SEEXTRCT for filtering, or if it first needs to be input to FSELECT.
- seplot - [P] Cyclically queries the user for additional SE files and then runs SEEXTRCT and FPLOT, for SOF.
- seselect - [P] Runs SEBITMASK and FSELECT to generate a FITS file.
- timetrans - [P] Translate relative time range to an absolute time range by reading XTE produced .lc files.
- trans2fits - Creates science event FITS file from Transparent Mode data.
- xenon2fits - Creates science event FITS file from Good Xenon data.
- xpcaarf - Creates a .arf file for an input XTE PCA .pha and .rmf file.
- xtederive - Second half of XTEFILT; uses output file from fcollect to derive additional useful quantities for XTEFilter file.
- xtefilt - [P] Creates a .xfl filter file for a given observation.
- xtetape - Read NASA/GSFC/ADF supplied XTE data distribution tapes.
- xteversion - [P] Print the version and date of the XTE sub-package.

Parameter File Manipulation tasks:

- pconfig - [P] Configure parameter files to user defaults
- pget - Get the value of one or more parameters in a parameter file
- plist - List parameters from one or more IRAF-style parameter files
- pquery - Get the value of a parameter in an IRAF-style parameter file
- pset - Set one or more parameters in an IRAF-style parameter file
- punlearn - Get a copy of the system parameter file

*** This tool is currently non-operational**

***** May not be supported at all sites**

[P] Perl script. You must have Perl installed to use this tool

B Notices of FTOOLS Releases

Notices of the FTOOLS Releases, online HTML versions of FTOOLS documentation (like this document), the official FAQ, and other goodies are available on the World Wide Web at:

`http://heasarc.gsfc.nasa.gov/docs/software/ftools/ftools_menu.html`

This is where the most up to date FTOOLS information is available, including the continually updated FAQ. Users should always check here first for answers to questions.
Notices of FTOOLS Releases, patches, and other bits of important information can also be obtained by an electronic mail exploder. The list is called **ftoolsnotes**. If you'd like to have your name added, send a message to `listserv@athena.gsfc.nasa.gov` requesting this action. This message should be sent from the account and host for which you would like to receive the notices. Leave the subject blank and include in the body of your request the following:

`subscribe ftoolsnotes` *name*

where *name* is replaced by your actual full name.

C Bug Reports

This is the fifteenth public release of FTOOLS. While all the tasks have been tested during development, bugs and conceptual oversights are inevitable in a project of this size. Users are requested to report bugs and suggested enhancements or new features for the FTOOLS package to the HEASARC group at NASA Goddard Space Flight Center by sending electronic mail to the following hotseat:

- `ftoolshelp@athena.gsfc.nasa.gov`

If you could add a descriptor to the subject field along the lines of **FTOOLS help** or **XSELECT help**, it will speed up the reply.