The Xselect User's Guide

Version 1.3

J. Ingham
August 6, 1994

ASCA Guest Observer Facility
Office for Guest Investigator Programs
Code 664
Goddard Space Flight Center, NASA
Greenbelt, Maryland, 20771
USA

Contents

1 Introduction **2**

1.1 Generalities 2

1.2 Aids to the User 4

1.3 Reserved names 5

1.4 Sessions 5

1.5 Data Entry 6

1.5.1 Obscats 6

1.5.2 Read 9

1.6 Xselect's files 11

1.7 Entering Filters 12

1.8 Making Selections 13

1.9 Saving clearing and setting 15

1.10 Extracting products Applying filters 15

1.11 Plotting 17

2 XPI **18**

2.1 Syntax 18

2.2 Special Characters 20

2.3 comments 21

2.4 XPI commands 21

2.4.1 alias 23

2.4.2 log 23

2.4.3 lparm 24

2.4.4 recall 24
2.4.5 scripts 25
2.4.6 ? 27
2.4.7 ?? 27
2.4.8 $- spawn or exit to shell 27

3 Xselect Commands **29**

3.1 Summary 29
3.2 Command Descriptions 30
3.2.1 bin 30
3.2.2 choose 31
3.2.3 clear 32
3.2.4 echo 35
3.2.5 exit 35
3.2.6 extract 35
3.2.7 faint 38
3.2.8 fast 38
3.2.9 filter 39
3.2.10 gisclean 46
3.2.11 help 47
3.2.12 hkbin 48
3.2.13 list 48
3.2.14 load 49
3.2.15 make 50

3.2.16 mkfbin 52
3.2.17 plot 52
3.2.18 quit 53
3.2.19 read 54
3.2.20 saoimage 55
3.2.21 save 56
3.2.22 select 58
3.2.23 set 61
3.2.24 show 65
3.2.25 sisclean 68
3.2.26 sispi 68
3.2.27 smooth 69
3.2.28 stop 70

4 Terminals **70**

5 List of Files **71**

Xselect

Xselect is a command line interface to the Ftools, for X-ray astrophysical analysis. It has four major functions:

- to organize the input of data through the observation catalogue, and store it internally for easy use.
- to enter intensity, phase, region, spectral, and timing filters, which will be applied to the event data.
- to make GTI by applying selection criteria to the Housekeeping and auxiliary filter data.
- to extract images, light curves and spectra from the event data, using the entered filters, as well as the GTI created by the applied selections.

Xselect also contains routines of importance for specific missions. Currently, there are only ASCA specific commands, namely sisclean, gisclean, faint and fast.

In addition, there are routines to plot up various aspects of the event data, and of the HK and auxiliary filter data.

There are also general utility commands to view the internal state of Xselect, to view the raw data files, to clear various selections, data choices, etc., and to save the products of the analysis.

In the Introduction section, we sketch the workings of Xselect. There is also a section on the command line interpreter used by Xselect: XPI. It allows scripting, aliasing, spawning, and command recall. It also has a few particularities that must be attended to.

In the Commands section, we list all the Xselect commands available, with a brief description. There is also a list of all the temporary files that Xselect produces, and each one's purpose. Finally, we point out some of the current shortcomings and features of Xselect.

A note on the usage of this manual before we proceed: each Xselect session is given a session prefix, which is prepended to the names of the product files, and appears in the prompts. In the rest of this manual, we will use the default session name: xsel. Replace this with your own session name for all the Xselect temporary files.

1 Introduction

This is a brief tutorial for Xselect. Only the general outlines of the analysis are sketched. For ASCA analysis, the ABC guide gives some suggested prescriptions, and modes of analysis.

1.1 Generalities

Xselect is a controlling shell for the Ftools. This means that rather than reading all events data into memory, Xselect holds the data as well as its temporary products in disk files, and processes them by piping these files through various ftools. Though this means more I/O, the core size of the program is kept fairly small.

One consequence of this method of operation is that Xselect performs markedly better if the data is stored on a disk that is directly connected to the cpu that is running Xselect, and can perform very badly over slow networks.

If you want to gain more insight into how Xselect uses the Ftools, and what temporary files are produced, use the ECHO command, which writes all the spawned commands back to the terminal before running them.

Xselect keeps track of five directories. The data directory is the place where the event data is stored. The HK files can be stored in a separate directory, the hk directory. There is a seperate directory for the auxillary filter files (MKF files), the MKF directory. Xselect will not alter any of these files, and can run when the user has only read access to these directories.

Xselect's scratch files are stored in the current working directory. This is also where the product files go, when they are saved. Finally, there is a directory for storing the user's observation catalogues, which initially defaults to the

current working directory. The user must have write access to both these directories for Xselect to function.

The first stage in the analysis is to set up Xselect's internal state for the particular mission-instrument-datamode combination you are going to study, and the directory wherein your data resides. This is done with the various SET commands.

The next step is to enter some data into the program. The observation catalogue (obscat) is the easiest way to do this. If you do not already have an obscat made, MAKE OBSCAT will create one for you. If you do already have one made, use the LOAD OBSCAT command to use it.

Then you use the CHOOSE command to read data from the obscat list into Xselect's internal list.

If the data needs some preliminary conversion, such as Faint to Bright conversion or Fast mode time corrections, this should be done next. These commands process the original files, and produce temporary files that Xselect will work on. If you want to return to the original data, you can CLEAR the conversion, and Xselect will use your originally chosen data. Again, Xselect will not alter your original data, and in fact will even work when you have only read access to the data directory.

Next, you may want to clean your data. There are three sets of commands that perform this function in Xselect. The first are the instrument specific cleaning methods, of which SISCLEAN for the ASCA SIS and GISCLEAN for the ASCA GIS instruments are the only examples.

The second set of commands are the FILTER commands. These commands do no work, they merely enter filters (region, timing, phase, pha_cutoff and intensity) that will be applied to the next run of the extract command.

The third set are the SELECT MKF, HK and EVENTS commands. These commands all work by applying some Boolean expression to one of the data files. These can be either the event, the HK or the MKF files.

In SELECT EVENTS, the events files are directly reduced, and Xselect will continue to use the products of the selection until you give the command CLEAR SELECTION. An example of this is selecting only one of the four chips of the ASCA SIS, which is accomplished with the command

```
xsel:ASCA- SIS1 > select event "CCDID==1"
```

In SELECT MKF and SELECT HK, GTI's are produced by applying the selection expression to parameters in the MKF or HK files, which GTI's will be applied on the next run of EXTRACT. More insight into the nature of these parameters can be gained using the MKFBIN and PLOT MKF commands to examine them.

Next, the filters are applied and the product files produced with the EXTRACT command. The possible products are images, spectra, and lightcurves. You can also output the event list that is the result of all the filtering and selections that were entered.

Finally, these product files can be PLOTted, and SAVEd for input into other analysis packages.

After dispensing with a few odd points, we will take up these topics in more detail.

1.2 Aids to the User

Xselect has several on line features to help the user. The first is the on line help. This is modeled after the VAX on line help, with topics and subtopics. To get help on a particular command, just type

```
xsel > help command
```

The only detail of note is that to get help on a compound command, you must surround the command name in double quotes to protect the spaces. For example:

```
xsel > help "show events"
```

will give you the "events" subtopic of the help topic "show".

If you type ??, you will get a list of all the available commands.

The next useful command is the XPI command LPARM. This will list all the parameters that a given command takes, as well as their types, and current default values. It is very useful for reminding yourself of what options any given command has.

All the commonly used parameters will be prompted for, if not entered on the command line. It is often best, when you are not sure of the order of parameters in a command, to just type the command name and let Xselect prompt you for the values it wants. Whenever a given command has a number of subcommands (e.g. SHOW STATUS, SHOW DATA,...), Xselect will list the available subcommands before giving the prompt.

On UNIX systems, Xselect uses the READLINE command line interpreter. This supports command line edition, filename completion (use the tab key, hit once it makes the longest unique completion, and if there is more than one completion, a second tab will show you all the possible options)and up and down arrow keys for command recall.

Finally, if you cannot remember the name of some file, or what exactly you called the directory where you stored your data, the $ command, with no arguments, will always break to the shell. You return to Xselect by typing exit, and you will come back at the point you left.

1.3 Reserved names

When Xselect exits, it deletes all files of the form (session name)_*.xsl and (session name)_*.tmp in the work directory, unless you save the session. Therefore, you cannot use names of this form, unless you are willing to have them deleted on exit.

1.4 Sessions

Xselect supports multiple sessions in the same directory, which can even run simultaneously, provided they have distinct names. On entry into Xselect, you will be prompted for a session name. Then all the temporary and product files written out by Xselect will have this session name as a prefix. The default session name is xsel. So for instance the light curve output

qdp file is called xsel_curve.xsl. You can also enter the session name when launching Xselect:

% xselect prefix=myown

will start up a session with the name myown.

When you exit Xselect, you are prompted whether you wish to save the session or not. If you respond yes, all the temporary files are saved, otherwise they are all removed. Do not erase any of these files on your own if you wish to restore the session at a later date, or Xselect will no longer be able to do so.

To restore a saved session, simply give your new session the same name as that of the session, and you will be prompted whether to restore the saved session. One caution, if you respond no to this prompt, the previous session will be cleared before Xselect proceeds and you will loose all the product files (except of course those you saved with your own filenames, with the SAVE command).

1.5 Data Entry

There are two ways to enter data into Xselect, by the observation catalogue (obscat), and the read command. We take these up in order.

1.5.1 Obscats

The obscat is a fits file containing event file names and other pertinent information about the files, such as on-time, datamode,..., gathered from keywords in the primary extension of the events files. The user can specify the items to be included in the obscat, and there are a set of built in defaults for the ASCA mission.

The data files must all be contained in the same directory, the data directory. The obscat is constructed by doing a directory listing on this directory, using a mission dependent regular expression for the file names. There is also the provision for user supplied regular expressions.

Although the catalogues are limited to the contents of one directory, the HK files can be kept in a separate directory.

Currently the MAKE OBSCAT command is designed primarily for the ASCA mission. It assumes the naming conventions of FRFREAD for the Event, HK and GTI files. However, the read command makes an obscat that can be SAVEd, LOADed and all the other functions the made catalogues admit.

If no observation catalogues are present, then one can be made by specifying the mission, the instrument, and the data directory (and optionally the datamode), then issuing the MAKE OBSCAT command:

```
xsel > set mission ASCA
xsel:ASCA > set inst sis1
xsel:ASCA-SIS1 > set datamode BRIGHT
xsel:ASCA-SIS1-BRIGHT > set datadir ../../data/20019000
xsel:ASCA-SIS1-BRIGHT > make obscat
```

The first step is unnecessary if you always use the same mission, since Xselect stores a default mission that it loads at startup. To change this default, you can start up xselect with the command

% xselect default_mission=ROSAT

This will cause Xselect to load the ROSAT mission at startup, rather than ASCA which is the default Xselect is shipped with.

A catalogue will be made in the obscat directory (or the current working directory if no obscat directory is set) with the default name xsel_(instru).cat, where (instru) is s0, s1, g2, g3 for the ASCA mission. It is saved when the Xselect session is exited, regardless of whether the rest of the session is saved.

When a catalogue is made, it is automatically loaded into xselect and the contents displayed.

To view the contents of the catalogue again, use the command

```
xsel >  show obscat
```

Note that the plain SHOW OBSCAT command will usually only show you a subset of the parameters that are in the obscat. To see all the parameters, record by record, use the command:

```
xsel > show obscat list
```

Or, if you have a different list of parameters that you want to view in the column form, use the command

```
xsel > show obscat displist="@myparam.list"
```

where myparam.list is a list of the parameters you want to view.

If you have already made some catalogues, you can get a listing of them with the command

```
xsel >  list obscat
```

This will list the catalogues in the current working directory (or obscat directory if one is set), and the data directory if one has been set.

You can load an already made catalogue with the command

```
xsel:ASCA >  load obscat
```

You can either specify, by name, the obscat that you want to load, or if you give no arguments Xselect will display the catalogues present in the obscat directory and the data directory, and prompt you for a choice.

This command will also set the datadir, mission, and instrument and data-mode (provided they are unique) from the catalogue, if you have not already done so.

To read data in from the catalogue, use the choose command. Choose works in two modes, you can say

```
xsel:ASCA >  choose 1-15
```

which will select the files 1-15 from the obscat. Or you can say

```
xsel:ASCA-SIS0 >  choose "datamode=='BRIGHT'"
```

which will select all the files from the obscat with BRIGHT datamode.

If you have a catalogue with the default name corresponding to the currently selected instrument, then choose and show will act on that catalogue automatically.

To apply selection criteria to the current catalogue to produce a new catalogue, rather than to just extract data from it, use the command

```
xsel >  select obscat  "datamode=='BRIGHT'"
```

This selection will not overwrite the previous catalogue, but it will rename it to move it out of the way. Since load obscat will give a brief description of each catalogue it finds, this should not be a problem, however if you want to save the current obscat with a name of your own choosing, you can do so with the command

```
xsel >  save obscat
```

Finally, the process of selection may result in many obscats that you don't want to keep. The CLEAR OBSCAT command will give you the same listing of the obscats as LIST and LOAD, and prompt you to remove those catalogues you no longer need.

1.5.2 Read

After all this, if you want to read in data without the aid of a catalogue, there is the command

```
xsel >  read filename
```

You can enter the files by hand on the command line, or dump their names to a file, say files.lis, and give @file.lis as the filename to the read command:

```
xsel > read "@file.lis"
```

The "s are necessary to keep XPI from expanding file.lis onto the command line (q.v. @ in XPI section).

There are several other functions which give more information about the data. To view the contents of the Event, or GTI files that have been entered, use the command

```
xsel >  show event,gti
```

To see a list of the data read in, use the command

```
xsel >  show data
```

To view the contents of the work or data directories, use the commands

```
xsel > list workdirectory
```

and

```
xsel > list datadirectory
```

Finally, to clear the data use

```
xsel >  clear data
```

1.6 Xselect's files

Once you have chosen the data, the list of files chosen is stored, along with the associated HK and GTI files, by Xselect. All further operations will act on these files, or on product files made in the course of analysis, where that is appropriate.

Xselect has one workspace for product files (e.g. from the FAINT or the SELECT EVENT commands). When this is filled by some command, say FAINT, then Xselect will point to the files in it, rather than the original files. However, the original file list is not lost, so by clearing the workspace (done by CLEAR FAINT, if the workspace was filled by FAINT, and similarly for the other commands, or as a last resort by CLEAR WORKSPACE), you get back to your original selection.

It is important to note that there is only one workspace, so if you give two commands that both use the workspace, for example a FAINT command, and then a SELECT EVENTS, the results of the FAINT command will be used in the SELECT command, but then will be overwritten. If you want to retain these files, then you should do the following:

```
xsel >  faint
xsel >  save faint
xsel >  select events
```

The saved workspace is duplicated, so that Xselect can continue to operate on the copy it keeps. The saved workspace files can be reloaded later, using the read command. Because the workspace is duplicated in the SAVE FAINT command, if you are short on disk space, it is a good practice to save these files, clear the workspace, and then read them in again using the READ command.

The commands that use the workspace are: FAINT, FAST, GISCLEAN, SELECT EVENTS, and SELECT FAST.

There are also some commands which produce a single product event file that is automatically used in future extract commands, again without losing the original data list. The two examples of this are the SISCLEAN command, and the EXTRACT EVENTS command.

When you save this filtered event list, you will be given the option to read it back into Xselect Doing this will clear the previous data, and reset the data directory to the current working directory.

Xselect also stores the extracted lightcurves, spectra and images until you decide to save them (e.g. with SAVE CURVE, ...). If you do not save them, however, they will be erased when you exit the session.

1.7 Entering Filters

Xselect supports region, phase and timing filters, as well as pha upper and lower bounds. These filters are not immediately applied to the events data, but stored and used in the next run of the extract command.

The filters are entered with the commands

```
xsel > filter time
```

```
xsel > filter region
```

```
xsel > filter phase
```

```
xsel > filter pha_cutoff
```

The timing filters can be entered in any of three formats, ASCII, or FITS GTI files, with start and stop columns, and Xronos window format files, with the command FILTER TIME FILE.

Timing filters can also be entered graphically using the FILTER TIME CURSOR command, after a light curve has been made.

Finally, you can enter GTI's by hand in either MJD, SpaceCraft Clock or UT with the commands FILTER TIME MJD, SCC, and UT.

The region filter is in SAOImage format. Region filters for the filter region command can be created graphically via the plot image command, which spawns an SAOIMAGE session. However the resultant regions have to be read back into Xselect by hand with the filter region command.

Another important point to note about the region filters, is that they contain no information about the physical image coordinates, only whatever binning you used in the image from which you made the region. So if you rebin your image, you must scale your region files accordingly before using them again.

At this point, only a lower and upper cutoff for the PHA are allowed.

The phase information is written into a Xronos-format window file (the sort created by the Xronos win command) by the FILTER PHASE command. You enter the epoch and period in days, then you can choose phase windows. Then Xselect will filter the events based on the union of all these phase windows.

You can save this phase information. To read it back into Xselect at a later time, use the FILTER TIME FILE command (which supports Xronos window timing files).

There are a few other filters created within Xselect which are automatically entered into the EXTRACT command. These are the GTI files (in FITS format) created by the SELECT MKF, and the SELECT HK commands, and the GTI files created in the selections made in the PLOT CURVE command.

The filters that have been entered can be listed by the command

```
xsel >  show filter
```

They can also be cleared by the various CLEAR commands (see clear under the Xselect_Commands subheading)

```
xsel >  clear region
```

...

1.8 Making Selections

The user can also apply selection criteria to the data itself, as well as the HK or auxiliary filter files, to filter the data. There are two provisions for selections based on data:

a) To apply a general selection expression based on the columns in the event files, use the command

```
xsel >  select event
```

This command produces a set of reduced events files, one for each input file, which are stored in the workspace. To save these files, use

```
xsel > save select
```

To aid in building up these expressions, the command

```
xsel >  show params event
```

will display the column names in the event list files. You can also use the command

```
xsel > show event
```

to look at the event data itself.

b) To apply a selection based on the columns of the auxiliary filter files, use the command

```
xsel >  select mkf
```

To see the columns available in the auxiliary files, use the command

```
xsel >  show parameters mkf
```

To gain insight into the behavior of these parameters, use the commands

```
xsel > mkfbin
```

```
xsel > plot mkf
```

The first will bin up a user supplied list of the mkf parameters. The second will plot up these binned curves. They can be plotted either against time, or against each other.

1.9 Saving clearing and setting

The SET, SHOW and CLEAR commands allow the user to control the parameters for EXTRACT and other commands.

The SET commands allow the user to set the bin sizes, and the column names for PHA, X and Y in the event list, which will be used in the EXTRACT command. You can also set the mission, instrument, datamode, as well as the plot device through the SET command. To restore the defaults for the missions currently supported by Xselect use the set mission command.

The clear command clears the various filters, and it can also clear product files, like those from FAINT and FAST, and return Xselect to the original data. There is currently no multistep undo, however; you can only return to the original list.

The SAVE command saves the various product files. Also, the SAVE SESSION command will save the state of the session on exit. This can then be restored at the start of the next session.

Finally the SHOW command is useful for displaying various elements of Xselect. In particular, the SHOW STATUS command gives a thumbnail sketch of the current state of Xselect. SHOW FILTER shows the currently entered filters. And, as mentioned above, the SHOW command also allows you to inspect the event and MKF files in your data.

1.10 Extracting products Applying filters

Extract is the command that actually extracts light curves, spectra and images from the event lists. Additionally, you can output the filtered events lists, which will speed up later analysis (this is not currently supported for ASCA GIS MPC data). Any combination of the possible outputs is allowed.

All the entered intensity, region, timing, pha and phase filters are automatically applied. The GTI's created by the SELECT MKF commands will be used. Also the binsize for light curves, and the rebinning factors for the spectra and the image, which have been entered with the SET command, are applied. If any product files have been made, for example the output files

from the select event commands, they will be used in place of the original files.

The output of the extract image command is an image in the primary extension of the FITS file xsel_image.xsl. It has all the keywords necessary for input into XIMAGE.

The output of the extract curve command is both an ascii QDP format, and a FITS format light curve file, xsel_curve.xsl. The former also contains, as comments, the accumulated spatial, timing and pha selections. This is primarily intended for a quick look, and to aid in further timing selection.

The output of the extract spec command is a spectrum, contained in the first extension of the FITS file xsel_hist.xsl, which also contains a weighted map image in the primary extension of the file. All the keywords necessary for XSPEC are written into the spectrum file.

The output of the extract events command is a FITS event list. The resultant GTI's are contained in the second extension of the file. This events list is suitable for input to the new, FITS-reading Xronos. However, if the input data to Xselect is not time ordered, this events list will not be either, so you will need to process it through the FMEMSORT ftool before passing it to Xronos.

As mentioned above, Xselect will continue to use this filtered event list in future work. If you want to save it, the command

```
xsel >  save events
```

The extractor also outputs an ascii and a Xronos window format file containing the resultant timing selections. The former can be viewed with the command

```
xsel >  show goodtime
```

These resultant timing selections, in ASCII and Xronos format, as well as the event list, the spectrum, light curve and image, can be saved with the save command.

1.11 Plotting

Xselect provides the facility to view the image (and to smooth it), the spectrum and the light curve output by the EXTRACT command, as well at to bin up and view the MKF parameters. Two of the plot commands, plot image and plot curve allow the graphical entry of selections.

The image most recently produced by the EXTRACT command can be viewed with the command

```
xsel >  plot image
```

This spawns an SAOImage session, and the user can select regions and write out the region description files. When we learn how to make Xselect talk with SAOImage, we will keep one SAOImage session open for the rest of the Xselect session, but for now, you will start up a new session with each plot image command.

If you want to smooth the image before viewing, use the SMOOTH command. Currently there are BOXCAR, GAUSSIAN and LORENTZIAN smoothing methods available.

The light curve most recently produced can be viewed with the command

```
xsel >  plot curve
```

This command also provides a mouse driven way to enter timing selections.

You can plot the spectrum with

```
xsel >  plot spectrum
```

Also, to aid in the SELECT MKF command, the columns in the MKF file can be plotted. This must be done in two steps, first bin them (more than one parameter can be binned at a time) using

```
xsel >  mkfbin
```

Then you plot the parameters with the command

```
xsel >  plot mkf
```

Again, you can see more than 1 parameter per plot.

2 XPI

This is the command line interface used in Xselect. It has some useful features for listing the available commands, their parameters, and the current defaults. It also has scripting and alias facilities. It is not case sensitive, and does command completion for both command names and parameter names, so you need enter only the shortest unique string for any command.

The UNIX version also inherits the nice command line editing features of the READLINE library, which has an EMACS like syntax for command line searches, and supports up and down arrow history recall, and filename completion using the tab key.

It has some magic characters, which are captured in the parser, and so they must be surrounded by double quotes to get past XPI.

In most cases, XPI is fairly forgiving, so you do not need too detailed a knowledge of it. For all practical purposes, you only need to know what the magic characters are (see magic), and the general syntax of the command line.

2.1 Syntax

The commands in Xselect consist of a command name, and a group of parameters. The parameters have two seperate qualifiers, they can be queried or hidden, and learned or fixed.

The queried parameters will be prompted for if not entered on the command line, and can be entered (provided the correct order is maintained) without their command name. The hidden parameters must be entered in the form

name=value and will not be prompted for. Rather a default is provided in the Xselect par file.

The learned parameters will have their defaults set by the last entered value, whereas the fixed parameters will revert to their original defaults when no value is specified for them.

The nature of any given parameter can be determined by the LPARM command.

The par file contains all the Xselect parameters, and their defaults. It will maintain the state of Xselect between invocations. There is a master copy of it that is kept with the Xselect executable, and this is copied into the users home directory the first time Xselect is used. This local copy is the one that will be used afterwards. If you want to reset the parameters in Xselect to the shipped state, just remove the par file from your home directory, and you will recieve a fresh copy the next time you start Xselect.

The convention for giving the syntax of commands that will be used in this help file is as follows:

```
xsel >  comm <par1name> <par2name> [par3name] [par4name]
```

Here, comm is the command, and parXname is the name of the parameter. The parameters in arrow-brackets <> will be prompted for if not given on the command line. The parameters in [] are hidden. Thus, in the above example, entering comm to the XSELECT prompt will result in the following:

```
xsel >  comm
> Value of par1name? >[default1]
> Value of par2name? >[default2]
xsel >
```

where the prompting messages are, of course, tailored to the situation and the user has to either enter the parameters when asked or type <CR> to get the defaults. To change the hidden parameters, the user must type

```
xsel > comm par3name=value3 par4name=value4
```

Order does not matter for hidden parameters.

All commands and parameter names can be abbreviated to the shortest unique string, and are not case sensitive. The same is true for the values of the verb_what parameters. I.e. the following all produce the same result:

```
xsel >  show obscat

xsel >  show obs

xsel >  sh OBS ...
```

2.2 Special Characters

XPI recognizes the following special characters:

```
! - indicates the rest of the line is a comment
@ - starts reading commands from a command file
$ - spawn to the system
" - protects other special characters
= - seperate parameter-value pairs
```

As a result, you should observe the following rules when entering expressions on the command line.

- An argument with imbedded spaces must be delimited by double quotes.
- An argument containing an @ or a ! must be surrounded by double quotes.
- = signs are trapped by XPI, so use .eq. in selection expressions, or surround your expressions with double quotes. In fact, to be safe it is always best to do this anyway.

2.3 comments

All text between a ! character and the end of line is treated as a comment. Comments are most useful when entered into a script. Since there is no difference between reading from the terminal or reading from a script, comments can also be entered interactively.

There are two types of comments. If the first character in the line is ! then the entire line is ignored and the next line is read. If ! is in the input line, but is not the first character, the ! and all following characters are removed. Thus if the software requires that you enter a blank line (i.e., default) and you wish to include a comment, then you must include one or more spaces before the !.

Examples:

```
xsel >  show       ! This is a comment
xsel >  ! this line is completely ignored
xsel >   ! this causes the program to read a blank line
```

2.4 XPI commands

Some of the commands in Xselect are really a function of the command line interpreter, XPI, and so are common with other HEASARC software like XSPEC and XIMAGE. These are generally commands that give information about the parameters used by Xselect, or do logging of sessions, history recall, etc.

Perhaps the most useful command of the lot is the LPARM command. LPARM Lists the PARameters that a given Xselect command will take, gives a short description of the parameter, and shows its current value.

XPI on UNIX systems uses the GNU READLINE library. This supports full command line editing (including EMACS style commands, e.g. Â for beginning of line Ê for end of line, and Ŝ for incremental search). The up and down arrow keys function for command recall. It does not support the full history mechanism in the READLINE library however, so !! will not

recall the previous command, for example. However, XPI also has a built in command recall that supports a similar syntax.

The following are the XPI commands:

- recall - Recall and replay previous commands
- alias - Alias complex commands to one word abbreviations
- script - Open and close script files
- log - Open and close log files
- lparm - lists the parameters for a given command
- ? - lists command groupings
- ?? - lists the commands available
- debug - enter debug mode, watch how the command line is parsed
- dumpkey - dumps the current copy of the key table
- dumppar - dumps the current (in memory) copy of the par file

The last three of these are really only for debugging

Along similar lines, but actually a part of Xselect, is that for most commands with known range of choices (set device, set instrument, and all the commands of the verb verb_what form), if you wait to be prompted (by hitting a return after the command name), you will be given a list of possible responses, and then prompted for your choice, e.g.:

```
xsel >  extract<ret>

ALL             --> extract a spectrum, image and light curve
CURVE           -->  extract a light curve
EVENTS          -->  output the filtered event list
IMAGE           -->  extract an image
SPECTRUM        -->  extract a spectrum
```

```
QUIT            -->  quit extract

> Give parameter to be binned >[all]
```

In the rest of the section we describe the XPI commands one by one.

2.4.1 alias

Alias a complicated command with a short one. For example:

```
xsel >  alias cleansis ‘‘sisclean clean_method=2‘‘
cleansis == sisclean clean_method=2
```

Note that the “ ” is necessary, otherwise XPI will think that clean_method is a parameter for alias, rather than part of the alias.

2.4.2 log

This command opens or closes a log file.

```
xsel >  log <log file>
```

where <log file> is the name of the file to be opened (default extension is .log). If no arguments are on the line, then the default file name is xselect.log. If <log file> matches the string “none” then the current log file is closed. The log contains a complete dump of the session – to just get the commands (for a macro, for example) use the scripts command.

Examples:

```
xsel >  log andy
```

will start logging commands in andy.log

Example 2:

```
xsel >  log none
```

will turn off the logging.

Xselect will log all the output it generates, as well as the output that is generated by the spawned Ftools. This feature is not yet available on the VMS version.

2.4.3 lparm

The lparm command_name command will give a list of the parameters for the given command, their default values, and a short description. This is very useful for reminding yourself of the command options.

2.4.4 recall

With no arguments, recall prints the last 20 commands given. With a numerical argument, it reruns that command from the command list. The only tricky part of this is that when you add a new commands the command numbers get shifted down by 1 for all the previous commands, so the same number will refer to different commands as you add more commands.

On Unix systems, this is no longer necessary, since the up and down cursor keys serve the same function.

Example:

```
xsel:ASCA-sis1 $>$ recall

      ...
    1 HELP
    2 SET inst sis1
    3 CHOOSE 1-2
    4 FAINT
    5 BIN image
    6 SAVE image -image.fits
    7 SHOW status
```

```
     8 HELP

xsel:ASCA-sis1 $>$ recall 3
xsel:ASCA-sis1 $>$ CHOOSE 1-2
```

2.4.5 scripts

This will cause XPI to open, close or use a script file.

At any prompt, it is possible to cause XPI to start reading from a script (indirect command file). For example,

Do you wish to continue? @YES.XCO

would cause XPI to search for the file YES.XCO. If the file is found the XPI would open the file, and read the answer from it. Thus the first line in YES.XCO should contain either yes or no.

To open a script file for keystroke recording, give the command

```
xsel >  script <script file>
```

where <script file> is the name of the file to be opened (default extension is .xco). If <script file> matches the string"none" then the current script file is closed. The script contains a list of the commands executed – to get a complete dump of the session use the log command.

Examples:

```
xsel >  script michelle
```

will start putting commands into michelle.xco

```
xsel >  script none
```

stops recording commands and closes the script file.

scripts search path If you use @filename to cause XPI to read from a script and it cannot find the script file in the current directory, XPI searches two other directories for the file. First, it checks to see if you have defined the environment variable MY_XCOMS. If MY_XCOMS is defined then XPI translates it and prepends the translation to the file name. Thus

setenv MY_XCOMS /users/my_name/my_xcoms

would cause XPI to search the directory /users/my_name/my_xcoms for the specified file.

scripts parameters It is possible to use parameters with scripts. To do this you enter the parameters on the line where you started the script. Thus,

```
xsel >  @test one two three
```

would cause XPI to open and read the test script using three parameters one, two, and three. When XPI is reading the script it will replace %n% with parameter n. Thus, %1% will be replaced with one, %2% with two, etc. The following illustrates a script that expects to be run with parameters:

make obs %1% choose %2% ...bin %3%

If you forget to enter the three parameters, then XPI will replace %n% with a null string.

It is possible for a script to call another script and pass in parameters. Thus,

@deeper first %2% %3%

is a valid line in a script. In this example, the first parameter is first, whereas the next two parameters will be set equal to parameters 2 and 3 of the current script. Also quotes can be used to denotes a single parameter with embedded spaces, or other magic characters. Thus,

@file This is all one" two three

would pass three parameters, and the first parameter would be the string This is all one.

scripts terminal prompt Sometimes it would be nice to write a script that in which most of the responses are read from the script, but one or more lines are read from the terminal. This can done by using the @ symbol with no filename. When XPI is reading an script, and comes to that line it will proceed to write the prompt on the terminal, and read the user's response. After this one line is read, XPI returns to reading the script. Of course, @ could again be used on the following line.

For example, consider the file:

```
1,23! Normal input
! Default input
@test! Read from the file test, until the end of the file
10 ! More input
@! Read this line from the terminal
999! Another normal line
```

2.4.6 ?

The ? command will give a list of the command groups available (command and admin for Xselect). Then typing command or admin will list the commands available in each group.

2.4.7 ??

The ?? command will list all available commands.

2.4.8 $ - spawn or exit to shell

Execute an operating system command.

```
xsel >  $ <command_line>
```

where command_line contains a UNIX command. No parsing of <command_line> is done by the XPI parser, therefore all the $'s, skipped fields, double quotes etc have the meaning they would in an operating system command. Also <command_line> must be 80 characters or less.

If the dollar sign is entered with no <command_line> the program opens a new shell, and a series of commands can be entered – type exit to get back into XSELECT. Spawning can be disabled by setting the environment variable DIASBLE_SPAWN.

Examples:

```
xsel > $
```

Spawns a shell.

```
xsel >  $ls
```

Lists the contents of the current directory.

```
xsel >  $cp triala0.pha x1822a0.pha;   ls *.pha;   exit
```

starts up a shell, copies a file and returns to XSELECT. The $ can even be given in reply to a prompt:

```
xsel >  set datadir
> Enter the event file directory > [] $
%ls
...
%exit
> Enter the event file directory > []
```

3 Xselect Commands

3.1 Summary

The following is a list of the Xselect commands that are currently available. For the XPI commands, see the XPI Commands heading.

- bin - Old name for extract
- choose - Choose observations from catalogue by number
- cpd - Equivalent to set device
- echo - Toggles echoing of Xselect command files to the screen
- extract - Extract image, spectrum or light curve from data
- filter - Set filters for the bin command
- faint - Do faint-to-bright conversion
- fast - Apply the SIS fast mode timing correction
- gisclean - Filter data using a PI-Rise Time window
- hkbin - Bin up an HK parameter(s)
- hksel - Create a GTI file using HK data
- list - List the obscats or the work and data directories
- load - Load an extant observation catalogue
- mkfbin - Bin up parameters from the Mkfilter output file
- make - Make observation catalogue
- plot - Plot spectrum, light curve, image, HK parameter, or other
- read - Read in filename(s) of FITS file(s) to be used
- save - Save current session, product FITS file, spectrum, light curve, image

- select - Apply Boolean expressions for data selection, for creating GTI intervals from HK or filter files, and for filtering the observation catalogue.
- set - Sets the mission, instrument and plot device, and the column names, and binsizes for extract.
- show - Show data files, obscat, parameters, selections, or status
- sisclean - Remove hot pixels from ASCA SIS data
- sispi - Fill the PI column for ASCA SIS data.
- clear - Clear input files, filters, selections, or everything
- echo - Toggles echoing of command files to the screen
- help - Obtain help on commands and syntax
- exit - Exit from Xselect
- quit - Exit from Xselect
- stop - Exit from Xselect

3.2 Command Descriptions

This section lists the commands and some of their most common options. Use the LPARM commands to view all the options for each command.

3.2.1 bin

Old name for extract

See extract.

3.2.2 choose

Choose data from Obscat

Choose enables the user to select data by number or selection expression from an observation catalogue (see MAKE OBSCAT). It is an alternative to typing in the full filenames in the read command. The call to CHOOSE clears the previously chosen or read in data, a READ called after a call to choose will add data to the list.

If you have read in filters, (e.g. by SELECT MKF, or FILTER ...commands) then you will be asked whether you want to clear the filters. If you say no, you will retain the filters, but the old data will still be cleared.

You can use a Boolean expression or a range as the argument to choose. If you give a Boolean expression, which must be based on columns in the currently loaded obscat, all those files matching the selection expression will be chosen. The selection expression follows the same syntax as in the select command.

If you give a range, then those rows in the obscat will be chosen. The syntax for the range is that commas delimit ranges, and dashes include all numbers within their termini. One element ranges are legal, and there is one special character: ** = last legal value.

Syntax:

```
xsel >  choose > <choose_which> <choose_clear>  [quiet]
```

Examples:

You can choose by position in the current obscat:

```
xsel >  choose  1-4,7,8-16,25-**
```

Or by a Boolean expression applied to the current obscat:

```
xsel >  choose  "EXPOSURE > 100"
```

3.2.3 clear

Clears variables, etc.

Clears various states of XSELECT.

Syntax:

```
xsel > clear clear_what
```

clear all Clears all data and logicals, and removes all the product files. Since this command removes all temporary files, you will be asked to confirm the clear.

Syntax:

```
xsel > clear all [proceed]
```

Example:

```
xsel > clear  all proceed=yes
```

This does a clear all, and does not prompt for confirmation.

clear clean Clears the region selection file produced by sisclean, and the cleaned events list.

Syntax:

```
xsel > clear clean
```

clear data Clears the read in data, but not the instrument name, plot device and other defaults.

clear datamode Clears the datamode, resetting it to 'NONE' which is its initial value.

clear events Deletes the output event list from extract, and points back to the data entered with the CHOOSE or READ commands. If you give an EXTRACT EVENTS, then Xselect will use the resultant events file in future computations, until you give the command CLEAR EVENTS.

clear faint Removes the products of the faint to bright conversion, and points back to the original data.

clear fast Removes the timing corrected files, and points back to the original data.

clear intensity Clears the current intensity selection. Again, you can choose 'all', or last, or give a file list. The files are all named xsel_intensity_gtinnn.xsl.

There is one slightly non-obvious point here: the following sequence

```
xsel > extract curve
xsel > filter intensity 0.5-0.9
xsel > extract all
...
xsel > clear intensity
xsel > filter intensity 1.0-1.5
```

will result in all the data being eliminated. The reason is that the clear intensity clears the GTI's, but the next filter intensity acts upon the lightcurve which was produced with these GTI's.

To produce the desired effect, instead do

```
xsel > extract curve
xsel > filter intensity 0.5-0.9
```

```
xsel > extract all
...
xsel > clear intensity
xsel > extract curve
xsel > filter intensity 1.0-1.5
```

clear hksel Removes the GTI files produced by the select hk command.

clear intensity Clears the intensity selection entered by filter intensity.

clear mkfsel Removes the GTI file produced by the select mkf command.

clear obscats This command physically removes obscats. When given with a filename argument, the named obscat is deleted. If no argument is given on the command line, you are given the same obscat listing as for LOAD or LIST OBSCAT, and then you can specify the obscats you want to remove by index from the listing.

clear pha_cutoff Clears the PHA lower and upper cutoff selections.

clear phase Clears the phase selection.

clear region Clears the spatial region filters. You can give a file list, or "all" or "last".

clear selection Removes the event list output by the select event command.

clear smooth Removes the smoothed image, so that the next call to plot image will show you the unsmoothed image. You do not have to clear the smoothed image to make another one, however. Smooth only acts on the unsmoothed image.

clear time This clears the various timing filters. The clear_what values are the same as the argument to filter time. All these commands will take 'last' and 'all' as well as a filename. The only ambiguity is that clear time file last clears the last Fits GTI entered.

clear workspace Clears the workspace, and unsets all relevant logicals.

3.2.4 echo

Echo ftools command lines

This toggles on and off the echoing to the screen of the command files spawned by Xselect.

3.2.5 exit

Exit Xselect

End the current XSELECT run and return to the UNIX operating system. Note that exiting will cause all the temporary files and filter information to be lost if they haven't been explicitly SAVE-d. You will be prompted whether you wish to save the current session.

Syntax:

```
xsel > exit <save_session>
```

3.2.6 extract

Extract light curve, image or spectrum

Extract product files from the entered data, using the entered filters and selections. Extract is the new name for the old bin command. Bin still works for the nonce.

The extract command works on the latest set of product event files that you have produced. So if you have run GISCLEAN, SISCLEAN, or SELECT EVENTS, extract will operate on the cleaned or selected files, not the original files. Similarly, if you have done a EXTRACT EVENTS, a later extract will use the output events list.

The possible products are a light curve (EXTRACT CURVE), in either QDP or FITS format (or both), a spectrum (EXTRACT SPEC), an image (EXTRACT IMAGE), or a filtered FITS events list (EXTRACT EVENTS). Any combination of these is also possible simultaneously. Also, EXTRACT ALL is equivalent to EXTRACT "CURVE IMAGE SPEC"

Extract assumes that there is a GTI extension, the extname is set in SET MISSION, in the same fits file which stores the event data. It accepts timing data in three forms, ASCII or FITS GTI form, and Xronos window format. It also accepts SAOIMAGE format region descriptor files, which are concatenated into one file called xsel_region.xsl, and fed to the extractor program.

If sisclean has been run, then the cleaned events list (stored in xsel_in_events.xsl) will be used. To clear this use clear sisclean.

If select hk or select mkf have been run, the resulting GTI's will be applied automatically. To clear this use clear hksel, or clear mkfsel.

If a timing selection has been made using plot curve, this too will be applied automatically.

The various bin sizes, etc, are inherited from the set command. Some of them can also be changed on the fly, by entering hidden parameters, but these values only apply to this invocation of extract.

Binned spectra, images and lightcurves can be plotted with the PLOT command.

Syntax:

```
xsel >  extract <bin_what> [use_qdp] [binsize_t] [pharebin_t]
               [xybinsize_t] [phalcut{\_}t] [phahcut{\_}t]
```

5 Examples:

a)

```
xsel >  extract ''CURVE EVENT"  use_qdp=no
```

Produces a light curve, and also a filtered event list. The use_qdp=yes flag causes the extract not to output QDP format light curves. To use this filtered event list for the next run of extract, say

```
xsel >  save event my_evtlist.fits yes
```

This is equivalent to saving the event list, and re-entering it using the read command.

b)

```
xsel >  extract ALL
```

this is equivalent to

```
xsel >  extract ''CURVE IMAGE SPECTRUM"
```

c)

```
xsel >  extract ''CURVE SPECT" bins = 30 pharebin = 4
```

Extracts a light curve and a spectrum, and temporarily resets the binsize and pha- rebinning factors.

d)

```
xsel >  extract ''CURVE SPECT" exposure=0.0
```

The **exposure** parameter controls the rejection of bins from the light curve (only), based on how much of the bin lies within the GTI's. If exposure=1, then a bin is only written to the light curve if it lies wholly within a GTI. If exposure=0, then bins are written to the light curve if any portion of them lies within a GTI. Exposure can have any value between these two extremes. Of course, if a bin does not fall inside any of the GTI's, it will not be written to the light curve, no matter what the value of exposure is.

```
xsel >  extract ''CURVE SPECT" bins = 30 pharebin = 4
```

3.2.7 faint

ASCA SIS faint to bright conversion

Perform faint-to-bright conversion on all files currently read into XSELECT. If the datamode is set to faint, then the EXTRACT command will run FAINT before extracting the spectra,..., provided of course you have not already done so.

For more details see the faint help page (i.e. from Xselect type $fhelp faint).

If you specify that you want to convert to bright2 mode, then you can either input (dfefile=your_filename), or have xselect create for you, (dfefile=MAKE) a Dark Frame Error file (see the help page for faintdfe for more details), which will be fed to the ftool faint. dfefile=NONE turns this option off.

If you set sispi=yes (the default) faint will fill the PI columns of the resultant BRIGHT or BRIGHT2 mode data files.

Syntax:

```
xsel >  faint <bright> <split> <dfefile> <echo> <maxgrade>
              [sis0echo] [sis1echo]  [sispi]
```

3.2.8 fast

Apply timing corrections to time column

ASCA fast mode data has no positional information, but the RAWY column contains corrections to the time column. The fast task applies these timing corrections to the time column of the file. The output is written to the workspace, and can be used in further calls to extract,...

To use fast, you need to have made an unbinned image in one of the other detectors, or in the detector you are analysing if there is some bright mode data available as well as the fast mode data. Fast will ask you for the X-position of the source, and for the instrument from which you got this information.

Syntax:

```
xsel > fast <x_image_center> <from_instrument> [save_file]
```

3.2.9 filter

Enter filters for extract command

This is the command for entering filters to be passed to EXTRACT. Each filter command builds on the list set up by the previous ones. For all the filter commands, a file name proceeded by a - indicates that that file should be removed from the list.

There are five filter_what values (besides quit), INTENSITY, TIME, REGION and PHA and PHASE, which will be treated in their own subheadings.

Syntax:

```
xsel >  filter <filter_what>
```

filter intensity After you have made a light curve, you can enter intensity selections, given as a range of acceptable values. The scale for intensity is read from the plot given in PLOT CURVE. The task converts this intensity selection to a set of GTI's, which are passed to the next EXTRACT run.

So

```
xsel > filter intensity .1-1.5
```

will PASS only those bins with an intensity between .1 and 1.5.

filter time This allows the entry of timing selections. There are three entry methods for timing filters, by the mouse cursor (FILTER TIME CURSOR), from a file (FILTER TIME FILE), or by the keyboard (FILTER TIME UT, SCC, MJD).

FILTER TIME FILE supports three formats ascii and fits GTI, and the Xronos window format. They can be used in any combinations, and the GTI forms allow multiple files. Xselect automatically senses the file type, so there is no need to specify this.

FILTER TIME CURSOR requires that a light curve has been made. You will be provided with instructions when you give the command on how to use the mode...

The other three commands (UT, SCC and MJD) are for entering the timing filters from the keyboard. The command names specify the timing system, Universal Time, Modified Julian Days, or SpaceCraft Clock. If a light curve has been made, then these commands will put up the curve, and you will see your selections as you enter them. These GTI's are then written to FITS files, which can be saved for later use with the SAVE TIME KEYBOARD command.

These timing filters are not immediately applied to the data, but rather stored for use in the next run of the extract command.

Some care must be taken in combining timing files. The way it is done is the following: The intervals within each file are ORed together, then all the separate input files are ANDed together to make up the resultant GTI file.

Consequently, if you wish to OR two ascii or fits GTI files, you have to put them together into one file. There is no requirement that they be time ordered, so for ascii files this is as simple as cat'ing the files. For FITS files the fmerge task will do the trick, or if you want to be fancier, the mgtime task. Xselect does not do this for you however (yet?).

To see that the extract comand has done what you expected, you can use the show goodtime command after you run extract to see the resultant GTI's.

Syntax:

```
xsel >  filter time file <file list>
```

Examples:

To enter two ascii files (ascii1.flt, and ascii2.flt), and two fits files (fits1.flt and fits2.flt) you would say:

```
xsel >  filter time file ''ascii1.flt ascii2.flt fits1.flt  fits2.flt"
```

To remove the file ascii1.flt, you can say

```
xsel >  filter time file ''-ascii1.flt"
```

Though CLEAR TIME FILE is more straightforward.

The list of files can be put in a file, say time.lis, one on a line, and entered via

```
xsel >  filter time file ''@time.lis"
```

The quotes are required (see the XPI magic subheading).

filter pha Currently you can only enter a pha lower and upper bound.

Syntax:

```
xsel >  filter pha <phalcut> <phahcut>
```

Example:

To restrict the extract command to 100 < PHA < 1000, say

```
xsel >  filter pha 100 1000
```

filter phase This allows you to enter one epoch-period combination, and up to ten phases. The subsequent extract command will use the union of all these phases to filter the data. If there is a phase file already present, you are given the chance to save it before proceeding.

XSELECT > filter phase epoch period phase <save_file>

The epoch is in MJD, the period in days, and the phase(s) run from 0-1.

So to filter using the epoch 40000, period of .5 days, and the first and third quarters of the period, say

XSELECT > filter 40000 .5 "0.0-.25,0.5-0.75"

The " "'s are necessary to protect the comma.

filter region Syntax:

```
xsel >  filter region <region>
```

This allows you to enter SAOIMAGE format region descriptor files. Multiple files are supported. The region parameter takes a list of files, which are added to the current list. If a file proceeded by a - in the list it is removed from the list. So to enter the regions reg1.flt, and reg2.flt, and remove the file reg.flt, give the command

```
xsel >  filter region ''reg1.flt reg2.flt -reg.flt"
```

Indirection with the @filename command is also allowed.

Some care must be taken in entering multiple regions, if you wish to get the result you expect. First off, note that the extractor takes in only one region file, so Xselect cat's them all together before passing them to extract. Since the order of the regions is important, you need to know that Xselect cats them in the order that they appear in the input list, appending the second file to the first, and the third to the end of these two, ...

To explain what the extractor does to these regions, here is an exerpt from the non- existent extractor help page on the subject (courtesy of B. O'Neel):

Summary: Each region specification is treated independently of the others. SAOimage has no syntax to specify AND/OR so each region stands on it's own. They are processed sequentially as the region file is read in, with the current region possibly overriding the previous selections. If the first region is an excluded region then the effect is to have an included region of the whole image inserted before any other region is processed.

Now, since that summary wasn't very understandable, let's do some examples with explanations.

Let's assume a 1000X1000 pixel image. 1 means the pixel is on, O means it is off.

Case 1:

An included box at 105,105 with a radius of 5,5. So, all the pixels

```
      (110,100)
  1111111111
  1111111111
  1111111111
  1111111111
  1111111111
  1111111111
  1111111111
  1111111111
  1111111111
  1111111111
 (100,100)
```

Now, if the same box was an excluded region then the exact opposite would happen.

```
       (110,100)
  0000000000
  0000000000
  0000000000
  0000000000
```

```
   0000000000
   0000000000
   0000000000
   0000000000
   0000000000
   0000000000
  (100,100)
```

with all the other pixels on.

An output from SISCLEAN is a collection of excluded points, so the image looks like...

```
   1111111111111111111111111111111111111111111111111111111111111111
   1111111111111111111111111111111111111111111111111111111111111111
   1111111111111111111111111111111111111111111111111111111111111111
   1111111111111111111111101111111111111111111111111111111111111111
   1101111111111111111111111111111111111111111111111111111111111111
   1111111111101111111111111111111111111111111111111111111111111111
   1111111111111111111111111111111111111111111111111111111111111111
   1111111111111111111111111111111111111111111111111111111111111111
   1111111111111111111111101111111111111111111111111111111111111111
   1111111111111111111111111101111111111111111111111111111111111111
   1111111111111111111111111111111111111111111111111111111111111111
   1111111111111111111111111111111111111101111111111111111111111111
   1111111111111111111111111111111111111101111111111111111111111111
```

What happens when two regions overlap?

Two boxes with corners overlapping. If they are both included then...

```
   0000000000000
   0111100000000
   0111111000000
   0111111000000
   0001111000000
   0001111000000
   0000000000000
```

The region file would look like

```
BOX(   )
BOX(   )
```

If they are both excluded then

```
1111111111111
1000011111111
1000000111111
1000000111111
1110000111111
1110000111111
1111111111111
```

The region file would look like

```
-BOX(   )
-BOX(   )
```

Now, if one is excluded and one is included, how should it look?

One way is

```
0000000000000
0111100000000
0110000000000
0110000000000
0000000000000
0000000000000
0000000000000
```

This means that the first box is the included box, and then there is an excluded box. The region file looks like

```
 BOX(  )
-BOX(  )
```

The other way is

```
1111111111111
1000011111111
1001111111111
1001111111111
1111111111111
1111111111111
1111111111111
```

This means that there is an excluded box, followed by an included box. The region file looks like

```
-BOX( )
 BOX( )
```

So, the order is significant. The overall way of looking at region files is that if the first region is an excluded region then a dummy included region of the whole detector is inserted in the front. Then each region specification as it is processed overrides any selections inside of that region specified by previous regions. Another way of thinking about this is that if a previous excluded region is completely inside of a subsequent included region the excluded region is ignored.

3.2.10 gisclean

Clean GIS data using RTI-PHA window

This task uses the RTI-PHA window contained in the calibration files rti_gis_nnnnn.fits, from the ftools refdata area to reject background events. You can also enter your own copy of the file with the parameter rti_table_user. The filtered output is placed in the workspace, and will be used in further calls to extract,...

Some ASCA data has the parameter RISEBINS=0. This data has no RISE TIME information, and you cannot run GISCLEAN on it. Xselect will warn you that you have this kind of data.

Also note that MPC mode data has no rise time information. So GISCLEAN is not appropriate for it either.

3.2.11 help

Get help on Xselect commands

Obtain help on the XSELECT commands, their syntax, and examples. Help uses the XHELP interactive help facility. The optional topic list can be used to go directly to a particular topic and sub-topic. Once in help, typing ?? will summarise extra features and special characters that can be used within the XHELP system.

To exit back to XSELECT, use an EOF, a /*, or type <CR> until you leave the help facility.

Syntax:

```
xsel >  help <topic list>
```

5 Examples:

```
xsel >  help
```

gives the list of topics available.

```
xsel >  help bin
```

gives the text for the bin command. This is a little funny, since bin is actually a subheading of help Commands. In fact any parameters entered on the command line after help point to the Command subheading, so to see the XPI section you must type help and wait for the XHELP prompt.

3.2.12 hkbin

Bin up HK parameters

Bins a curve of selected parameters from an HK file. If the files are in unexpanded form, they will be expanded as well.

Syntax:

```
xsel >  hkbin <param> <dtime>
```

5 Example:

```
xsel >  hkbin ''RBM_MON GASX" 120.0
```

This bins the relevant parameters in 120-sec bins.

3.2.13 list

List the extant obscats, work and data directories

Lists the obscats in the obscat directory, and in the data directory, if one has been set. Or the files in these directories.

The other parameters control the display for list obscat.

list obscat Syntax:

```
xsel >  list <list_what> [brief] [all] [only_datadir]
```

If brief is yes, a brief description of each obscat will be given.

If all is yes, then all obscats in the obscat and data directories will be listed. Otherwise, only those from the currently set mission, instrument (if set) and data directory (if set) will be listed.

If only_datadir is yes, only the data directory obscats will be listed.

Example:

This will display a full listing of the ASCA SIS0 catalogues in the obscat directory:

```
xsel > set mission ASCA
```

```
xsel:ASCA > set instrument sis0
```

```
xsel:ASCA-SIS0 > list obscat brief=no all=no
```

list datadir Lists the files in the set data directory

list workdir Lists the files in the current working directory

3.2.14 load

Load an extant obscat

Syntax:

```
xsel >  load <load_what> <cat_name> [brief] [all] [only_datadir]
```

Loads an externally saved catalogue. This catalogue is then used for choose, select obs and show obscat.

The command load obscat, with no catalogue name argument, will display a list of the catalogues in the obscat directory, and the data directory if one is set. The user can then choose from this list. Xselect will be automatically set for the mission, instrument and data directory of the loaded catalogue.

Otherwise the named obscat will be loaded.

The other parameters control the search for obscats, and the display. They work the same as for list obscat.

5 Example:

This loads the catalogue her_x1_sis0.cat:

```
xsel > load obscat her_x1_sis0.cat
```

3.2.15 make

Make an obscat

Makes an observation catalogue for the currently set ASCA instrument. If no instrument is set, you will be prompted for one. You can keep one catalogue per instrument per session prefix, and they can be accessed by switching the instrument name.

(ASCA is the only mission currently supported by make obs.) The catalogue is composed of the values, from the primary header of each specified file, of the keywords listed in the parameter obslis (or in the file filename if obslis has the value @filename). For ASCA the obscat includes all the 'Modal' keywords present in the event files.

Make can also use a selection critereon on the keywords in obslis to filter the catalogue as it is made. This selection string is stored in the parameter cat_filt, and follows the same syntax as in the select command. If cat_filt = DEF then the default expression is used, which is, for the SIS,

EXPOSURE>100&&NEVENTS>0

and for the GIS,

EXPOSURE>100&&NEVENTS>0&&HV_RED=='OFF'

If a datamode has been set, then the default filter will also include the datamode.

This observation catalogue is not deleted on exit from XSELECT. It is the user's responsibility to remake the catalogue if the contents of the data directory change. This means that you do not have to remake the catalogue every time you enter Xselect. Further if it is present, the choose and show obscat commands will act on the catalogue specified by the session prefix, and instrument name.

The user option for the make_inst allows the user to enter his/her own list string with which the catalogue will be created. The format is the same as for the ls command in UNIX.

You do not have to be in the directory which contains the data. You will be prompted for the directories containing the Event data. Further, if you end the directory string with a ?, e.g.

```
xsel > make obs '' /asca/data? "
```

then you will be shown an ls of this directory and prompted for the data directory again.

The catalogue is automatically displayed once it is made. However, this feature can be toggled on and off with the command

```
xsel > set DUMPCAT
```

This is useful for scripts, since the prompt to page the catalogue is an Ftool prompt, and so cannot be read from the script file.

Additionally, the catalogues can be viewed using show obscat. Another parameter, displist, lists the columns to be displayed in both cases. Both these parameters understand the @filename syntax for indirection.

The point of making obslist and displist seperate parameters is that it may be convenient to put more information in the catalogue than you want to see listed.

Data files can be selected from the observation catalogue by number using the choose command, as an alternative to using read and typing in the filenames.

Further, the catalogue does not need to be remade from session to session, but can be accessed straightaway using choose.

Syntax:

```
xsel >  make <make_what> <cat_filt> <data_dir> <make_inst>
```

5 Examples:

```
xsel >  make obs ''/asca/data/100010"
```

```
xsel >  make obs ''/asca/data/100010" sis0
```

3.2.16 mkfbin

Bin up auxillary filter files

Bins a curve of selected parameters from an Mkfilter output file.

```
xsel >  mkfbin <param> <dtime>
```

If no value of mkf_name is given, then Xselect will look for the mkf file in data_dir, using the regular expression in the hidden parameter mkf_def_expr. Otherwise it will use the file given.

```
xsel >  mkfbin ''RBM_MON GASX" 120.0
```

This bins the relevant parameters in 120-sec bins.

3.2.17 plot

Plot light curve, images, spectra,...

Plots spectra, light curves, images, HK and MKF parameters.

```
xsel >  plot <plot_what>
```

Plot curve also allows the user to enter timing selections graphically. Plot image allows you to construct region files.

plot curve This will put up a light curve. You will be asked if you want to enter timing selections. This works straightforwardly with an Xwindows system. For some terminal emulators, you must be careful that the GIN termination is set to null. I know this is true of VersaTerm Pro on the Mac.

If you choose to enter selection regions, a description of the available commands can be shown before you proceed.

plot HK Plots the HK parameters. These parameters must first be binned with the hkbin command. You can plot any or all of these parameters. By specifying the hidden parameter curves_per_plot, you can change the number of curves shown on each plot. 8 is the maximum, but is pretty hard to read. 5 is quite legible.

plot image Plots the accumulated image by spawning SAOImage. You can also construct SAOImage regions here.

plot MKF Plots the MKF parameters. These parameters must first be binned with the mkfbin command. You can plot any or all of these parameters. By specifying the hidden parameter curves_per_plot, you can change the number of curves shown on each plot. 8 is the maximum, but is pretty hard to read. 5 is quite legible.

plot spectrum Plots a spectrum.

3.2.18 quit

Quit Xselect session

End the current XSELECT run and return to the UNIX operating system.

Note that exiting will cause all the temporary files and filter information to be lost if they haven't been explicitly SAVE-d.

3.2.19 read

Enter datafiles by name

Read in up to 999 datafiles.

```
xsel >  read <readmode> <infiles> <usegti> <hkfiles> <expand>

xsel >  read science ''file1 file2" yes

xsel >  read hk hkfiles=''file1hk file2hk" expand=no
```

The possible values of readmode are:

- e - reads in the Event (science) files, and if usegti=yes, the gti files, using a default naming convention.
- hk - reads in the HK files. The variable expand tells whether they are in expanded form or not.
- d - reads in the Event files, and the gti and HK files uning a default naming convention.

Two subsequent reads, with no clear in between, adds the new data files to the list of files already entered.

You do not have to be in the directory which contains the data. You will be prompted for the directories containing the Event and HK data.

Subsequent read commands add to the file list, unlike choose, where the list is automatically cleared before each choose.

Read will attempt to get the mmission, instrument and datamode from the entered files. It is an error to enter files with different instruments, missions or datamodes.

It is often easier to respond to the prompts; see the example.

5 Examples:

Example 1:

```
xsel >  read d ft930317_1753.1614G30470M.fits

> The Event file directory > [/tmp/alan]
> Are GTI extensions available? >[yes]
> The HK file directory > [/tmp/alan]
> Are the HK files in expanded form? >[] yes
```

Example 2:

```
xsel >  read hk

> Give name[s] of FITS HK files >[] file1hk.fits file2hk.fits
> Are the HK files in expanded form? >[] yes
```

This reads in the hk files file1hk.fits and file2hk.fits, and informs the program that they are in expanded form.

3.2.20 saoimage

Old name for plot image

Start up SAOIMAGE, which will automatically read in the last image file accumulated using the image command. Control returns to the XSELECT program after saoimage is spawned.

```
xsel >  saoimage
```

3.2.21 save

Save product files...

Save a light curve, a spectrum, or a filtered list of the original type. Save session saves the current state of the session, and exits Xselect.

```
xsel >  save <save_what> <outfile>
```

save all Saves the cleaned events list (or the filtered events list if none), and any of the IMAGE, CURVE, or SPECTRUM which have been made. One base name is given, say xsel_save, then you get

- xsel_save.evt – the events list
- xsel_save.lc – the light curve
- xsel_save.pha – the spectrum
- xsel_save.img – the image

save clean Saves the events list, and region file or hot pixel list from sisclean (q.v.).

save curve saves the lightcurve

save dfe Saves the Dark Frame Error history file made with FAINT.

save events Saves the event list output by bin event. You will also be prompted whether you want to re-enter the event list. If you answer yes, the data and selections will be cleared, and the new output event list entered into the filename list. It is assumed for now that the output event list has been placed in the current working directory.

save expanded Saves an expanded HK file.

save faint saves the workspace files created by running faint.

save fast saves the workspace files created by running fast

save goodtime Save the goodtime intervals produced by the most recent run of extract.

save hksel Saves the output GTI from running select hk.

save image Saves the image file. You can also save the smoothed image, if one is made, with the command

```
xsel > save image save_smooth=yes
```

save intensity Saves the intensity selection GTI's from FILTER INTENSITY

save mkf_sel Saves the output GTI from running select mkf.

save obscat saves the current obscat with a user specified name

save phase save the current phase information in a Xronos window format file

save region saves the accumulated region file.

save selection saves the workspace files created by a select events command

save session Saves all the workspace files, and the current state of the program, and exits xselect. When you restart, you will be prompted as to whether you want to restore the session.

save spectrum Saves the spectrum file. In addition, the output spectrum can be grouped or rebinned on output. The default is to rebin the spectrum. This can be turned off by saying

```
xsel > save spec group=no
```

Otherwise, the following will be done:

```
MISSION  INSTRUMENT DATAMODE  ACTION
=======  ========== ========  ========================================
ASCA        GIS      PH       GRPPHA: GROUP channels 0-1023 by 4
            SIS      BRIGHT   RBNPHA: BRIGHT2LINEAR to 512
                              GRPPHA: Channels < EVTR flagged bad
            SIS      BRIGHT2  RBNPHA: LINEAR to 1024
                              GRPPHA: Channels < EVTR flagged bad
ROSAT      PSPC      NONE     GRPPHA: Bad 1-11, 212-256
                              GRPPHA: GROUP 12-211 by 10
```

save time SAVE TIME CURSOR will save the cursor timing file mades with FILTER TIME CURSOR, and SAVE TIME KEYBOARD will save the GTI files made with FILTER TIME UT, SCC, or MJD.

save workspace Saves the workspace files, whatever their origin.

3.2.22 select

Select data using event, hk or filter files

Select the event lists, the HK files, the MKF files or the obscat on the basis of any available criteria based on the columns in the FITS file that can be encoded into a Boolean expression.

```
xsel >  select <select_what> <expression> [mkf_name] [mkf_def_expr]
```

The following guidelines on how to construct the <expression> are taken directly from the help file for the FSELECT FTOOL:

This task creates a new table from a subset of rows in an input table. The rows are selected on the basis of a boolean expression whose variables are table column names. The column names are not case sensitive, but the values are. If, after substituting the values associated with a particular row into the column name variables, the expression evaluates to true, that row is included in the output table.

For more details on the syntax for the Boolean expressions, see the help page for fselect, (i.e. from within Xselect type

```
xsel > $fhelp fselect
```

select event Select data from the event lists.

Syntax:

```
xsel > select event <event_sel>
```

Examples:

To select data from within a circular region centered on (X,Y)=(125,125) with a radius of 10.0..

```
xsel >  select event ''Sqrt((X-125)\^2 + (Y-125)\^2) < 10.0"
```

Note that for region selection and timing selection, it is quicker to pass the criteria as filters to the extract command.

select fast Selects either the region INSIDE or OUTSIDE the mask region in ASCA SIS fast mode data. This is only available for fast mode data that has had the area mask disabled, i.e., Sn_ARENA=0.

Syntax:

```
xsel > select fast <in_or_out>
```

select HK Create a GTI file from criteria applied to data in the HK files. This file is automatically applied to the next running of extract. It can be saved with save hksel.

Syntax:

```
xsel > select hk  <hk_sel>
```

select MKF Create a GTI file from criteria applied to data in the MKF file. The MKF file is found by using the default mkf expression, mkf_def_expr, which can be an ls style regular expression, on the contents of the current data directory. Alternately, you can specify the mkf file name, and directory in the parameters mkf_name and mkf_dir. mkf_name must be a literal file name, not a regular expression. You cannot currently use the mkf_dir and mkf_def_expr to specify the file name . This is a bug that will be fixed in the next release.

This file is automatically applied to the next running of extract. It can be saved with save mkf_sel.

More than one file is allowed.

Syntax:

```
xsel > select mkf  <mkf_sel> [mkf_dir] [mkf_name] [mkf_def_expr]
```

Example:

```
xsel > select mkf ''SAA.eq.0''
```

This makes a GTI from the mkf files given by the mkf_def_expr in the current data directory.

```
xsel >  select mkf ''ELV_MIN.gt.10." mkf_name=mkffile.mine  mkf_dir=./
```

This makes a GTI from the mkf file mkffile.mine in the current directory.

select obscat Syntax:

This creates a reduced catalogue, based on a selection expression. The original catalogue is moved to a new name. If the original catalogue was xsel_s1.cat, then it will be moved to xsel_s1_v.cat. The new obscat will have the name xsel_s1.cat, and will be automatically loaded for you. So choose and show obscat will now act on this catalogue. N.B. this command does not read in any data, it only produces a reduced catalogue. You will now have to choose data from it.

```
xsel > select obscat <cat_sel>
```

Example:

```
xsel > select obscat ''datamode=='BRIGHT'''
```

This makes a reduced obscat containing only files for which the datamode is BRIGHT.

3.2.23 set

Set Xselect parameters

Sets various default parameters.

Syntax:

```
xsel >  set <set_what>
```

set binsize Sets the time binsize.

Syntax:

```
xsel >  set binsize <binsize>
```

set datadir Sets the data directory.

Syntax:

```
xsel >  set datadir <datadir>
```

Examples:

```
xsel > set datadir ../../data/20019000
```

Sets the datadirectory to ../../data/20019000

If you put a '?' at the end of the directory path, Xselect will do an ls on the directory, and prompt you for the directory again:

```
xsel > set datadir ../../data?
20019000  40001000
which data directory [../../data?] ../../data/20019000
xsel >
```

set datamode This sets the datamode. This has several effects. Some datamodes need special settings, (e.g. ASCA SIS fast, or ASCA GIS MPC mode), which are not set by set instrument. If a datamode is set, make obscat will filter on datamode. The datamode controls the grouping of spectra in the save spectrum command. Also, Xselect can do the appropriate preliminary analysis if the datamode is correctly set. For instance, if you read in FAINT mode data, the EXTRACT command will now run FAINT first, if you have not done so already...

This being said, except for filtering the obscat, it is not necessary to set the datamode yourself, since Xselect will set it when you READ or CHOOSE data files.

set device Sets the plotting device.

See help Terminals for more information about plotting devices.

Syntax:

```
xsel >  set device <plotdev>
```

set dumpcat Toggles on and off the display of the catalogue in the show obscat command. This is useful in scripts.

set image Sets the image coordinates to SKY or DETECTOR coordinates. This is a mission independant version of the set xyname command.

Example:

```
xsel> set image sky
```

is equivalent to

```
xsel > set xyname X Y
```

```
xsel> set image detector
```

is equivalent to

```
xsel > set xyname DETX DETY
```

set instrument Sets the instrument used in constructing and accessing the observation catalogue

Syntax:

```
xsel >  set instrument <set_instru>
```

Example:

```
xsel > set inst sis0
```

This sets the instrument to sis0. If you have forgotten the names of the available instruments, just say set instruments without an argument:

```
xsel > set instru
The available instruments are:
     SIS0
     SIS1
     GIS2
     GIS3

which instru? []
```

set mission To change between missions – the FITS file keywords are different for different missions, thus the mission has to be explicitly set. Currently supported missions are Astro-D (ASCA; default), BBXRT, and the Einstein IPC. Use the mission command before the read command.

set mkfdir This sets the directory in which Xselect will look for the MKF files.

There is one convenience that Xselect offers that may be confusing if you are not aware of it...When the data directory is set, if no MKF directory has been set, Xselect will look in the data directory for files matching the mkf default expression, and if it finds them, it will set the mkfdir to the data directory. If there are no MKF files in that directory, then Xselect will look in the directory datadir/../aux (this is the shipped value of the parameter mkf_reldir). If it finds mkf files in that directory, it will set mkfdir to that. Otherwise mkfdir will no be set.

However, this only happens if mkfdir has not previously been set. This search will not override a value set either with the SET MKFDIR command, or through a previous application of this search.

set pagewidth Sets the pagewidth for show obs.

set pharebin Sets the pharebinning factor.

set phaname Sets the column name for the energy values.

set wmapname Sets the coordinate used for the weighted map in the primary extension of the spectral file.

set xybinsize Sets the rebinning factor for the image.

set xyname Sets the column names used for accumulating the image inthe extract command.

3.2.24 show

Display Xselect settings

Show current settings or values for data, parameters, selections or general status.

```
xsel >  show <show_what>
```

show data Which data files are we currently working on? If chosen from the catalogue, the place in the catalogue is listed in the INDEX column.

show diff Syntax:

```
xsel show diff <compare> [copyover]
```

Displays the obscat in diff form (see fhelp fcatdiff). The parameter compare gives the columns in the obscat to compare, and the copyover parameter gives the columns to copy over to the diff display.

Example:

show diff compare="RAWXBINS TIMEBINS RISE_BINS"

show event Displays the event extension of a data file. This can be referenced either by its number in the observation catalogue (show_from = o), or by its place in the list of read-in data files (show_from = d). Use show data to determine the latter. The number is given in the parameter show_which. This parameter currently accepts only one entry, and not a range of values.

The show_rows parameter controls which of the rows you wish to see. The show_method determines whether the data is displayed by column (show_method =dump), or by record (show_method=list).

Syntax:

```
xsel > show event <show_from> <show_which> [show_rows] [show_method]
```

Example:

So to show rows 1-50 of the 5th file from the current obscat, say

XSELECT > show event obscat 5 show_rows=1-50

show filters Displays a list of the filters and selections currently loaded.

show goodtime Prints to the screen the resultant timing selections from the previous running of extract.

show GTI Displays the GTI extension from the data files. Same parameters as show events.

show MKF Displays the first extension of the current MKF file. The row and method options are the same as for show event.

Syntax:

```
xsel > show MKF <show_from> <show_which> [show_rows] [show_method]
```

show obscat Display the observation catalogue. The columns displayed are given in the parameter displist, and can be listed, one to a line, in the file filename if displist = @filename. If displist is blank, the display list defaults to:

```
INSTRUME
OBJECT
DATAMODE
BIT_RATE
DATE-OBS
TIME-OBS
NEVENTS
EXPOSURE
```

The show_rows and show_method parameters work as for show event.

show parameters Lists the columns in the current event, mkf or obscat FITS file. Useful for determining the arguments to use for the SELECT command.

Syntax:

```
xsel > show obscat [show_rows] [show_method]
```

show primary Displays the primary header of the events files. Files can be chosen either from the obscat or from the currently entered data.

Syntax:

```
xsel > show primary <show_from> <show_which>
```

show select Show Boolean statement describing most recent selections.

show status A thumbnail sketch of the status of Xselect, includes the information from show filter, as well as data information and default settings.

3.2.25 sisclean

Remove hot pixels from ASCA SIS data

Runs one of two methods to clean the image of hot and flickering pixels. Specify the method with the clean_method parameter.

Method 1) This removes pixels by a straight cutoff, all pixels with more than n photons per pixel will be eliminated. First, an image is accumulated. Then, to aid the eye, a Gendreaux plot is displayed. If saoimage is yes, then the uncleaned image is displayed as well. Then a filtered events list is produced. This will be used in further processing, and can be saved by the save clean command.

Method 2) This method uses a Poissionian distribution to reject hot and flickering pixels. For more details, see the help file for cleansis (e.g. give the command fhelp cleansis). First a filtered event list is produced, then this is fed to cleansis, a final event list is produced, which has had all the hot pixels removed. Also a list of the hot pixels is written out. The cleaned file and the hot pixel list can be saved with the save clean command. The uncleaned events file can be saved with the save event command.

3.2.26 sispi

Fill PI column for ASCA SIS data

This uses the FTool sispi to fill the PI column for ASCA SIS data. See the fhelp page for more details.

Syntax:

```
xsel > sispi [pha2pi]
```

The parameter pha2pi gives the calibration file. The value REF (the default) will find the appropriate file in the FTools refdata area.

3.2.27 smooth

Smooth the images

Currently, GAUSSIAN, BOXCAR and LORENTZIAN smoothings are permitted. The original image is preserved, and a new, smoothed image is written. Future calls to PLOT IMAGE will view the smoothed image (unless you CLEAR SMOOTH). Future calls to smooth, however, will smooth the original image again, and overwrite the previous smoothed image.

The output image will preserve the data type of the original image, unless you set the parameter smooth_outtype.

Syntax:

```
xsel > smooth <smooth_method> [smooth_outtype]
```

smooth gaussian You give the sigma, and optionally the nsigma for the gaussian. For more details, see the fgauss ftools help page.

```
xsel > smooth gauss 1.5 nsigma = 2
```

smooth lorentzian You give the sigma, and optionally the nsigma for the lorentzian. For more information, see the florentz ftools help page.

```
xsel > smooth lorentz 1.5 nsigma = 2
```

smooth boxcar Give the x and y sizes for the boxcar. For more information, see the fboxcar ftools help page.

```
xsel > smooth boxcar 3 3
```

3.2.28 stop

Exit Xselect

End the current XSELECT run and return to the UNIX operating system.

Syntax:

```
xsel > stop <save_session>
```

Examples:

```
xsel >  stop  save_session=no
```

exits the program, without saving the session. quit and exit do the same thing.

Note that exiting will cause all the temporary files and filter information to be lost if they haven't been explicitly SAVE-d.

4 Terminals

XSELECT will currently run and produce graphics on the following device types:

```
GF  (GraphOn)  GTERM  NULL  (!)  PS    RETRO  SUN
TEK4010V  PS  XTERM  XWINDOW  X11  GTERM   VT125O  MAC
```

Thus, Sunview users should be operating in a GTERM or SUN window. Set the plotting device using the DEVICE command, or by setting the "plotdev" variable in the xsel.par file to "/xterm" (for example).

When the plotting device is changed the new value is stored as the default in the xselect par file. On startup, if there is a saved session, Xselect first

looks for the plotting device in the save session file (xsel_session.xsl). If it is empty there, or there was no saved session, Xselect looks in the par file for this device. If this parameter is empty, then Xselect checks the environment variable PGPLOT_TYPE, and uses its value if found. If no plot device was found, you will be prompted for one the first time it is needed. If you respond with a ?, then you will get a listing of available device types.

To change plot device WHILE IN PLT, you can type e.g. device /gt, then o or p will redisplay the plot on the new device. Note that on the Macs /tek seems to work more quickly and with less screen-flashing than /mac.

Note also, that SOAIMAGE will not run on TEK, RETRO, GF, or MAC terminals.

2 Known_Bugs/Features

- The code explicitely assumes that there is one events extension per file. Future decisions to pack lots of extensions into a single FITS file are not yet supported!
- For the obscat to work, the relevant parameters must be stored as keywords in the primary extensions of the events files.
- The method of storing HK parameters for ASCA has changed, so that now all the HK data for the whole sequence is stored in one file. Xselect does not support this format automatically, so to view the HK data, you must read it in by hand using the read hk command.

5 List of Files

This is a list of the temporary files that Xselect produces as it runs.

- xsel_hist.xsl – temporary file for spectral data
- xsel_qdp_curve.xsl – temporary file for the QDP light curve data
- xsel_fits_curve.xsl – temporary file for the FITS light curve

- xsel_image.xsl – temporary file for the image
- xsel_region.xsl – input region file for EXTRACT
- xsel_fits_in.xsl – list of input FITS GTI files for EXTRACT
- xsel_mkf_out.xsl – fits timing file from the SELECT MKF
- xsel_ascii_in.xsl – input ascii timing file for EXTRACT
- xsel_ascii_out.xsl – cumulative timing file, computed by EXTRACT
- xsel_xronos_in.xsl – input Xronos window file for EXTRACT
- xsel_xronos_phase.xsl – xronos window file containing phase selections
- xsel_xronos_out.xsl – output Xronos window file compputed by EXTRACT
- xsel_hksel_out.xsl – is the FITS timing file from SELECT HK
- xsel_in_event.xsl – output events file from SISCLEAN, overrides other input to EXTRACT if present
- xsel_out_event.xsl – output events file from EXTRACT
- xsel_merged.xsl – temporary file for a merged event dataset.
- xsel_mergti.xsl – temporary file for the GTI associated with xsel_merged.xsl
- xsel_merghk.xsl – temporary file for the merged HK files
- xsel_work1 – root name for events workspace 1
- xsel_work2 – root name for events workspace 2
- xsel_workgti1 – root name for GTI workspace 1
- xsel_workgti2 – root name for GTI workspace 2
- xsel_workhk1 – root name for HK workspace 1
- xsel_workhk2 – root name for HK workspace 2
- xsel_ffcurve.fits – MKF curve file from MKFBIN
- xsel_mermkf.xsl – the temporary merged MKF file

- xsel_hkcurve.fits – the HK curve file from HKBIN
- xsel_cursor_time – root name for the cursor timing files
- xsel_temp.xsl – general purpose temp file
- xsel_session.xsl – stores the saved session data
- xsel_xsel.run – script file for the spawned Ftools
- xsel_files.tmp – temp file for file lists
- xsel_obscat.tmp – temporary file used for creating the obscat
- xsel_read_cat.xsl – temporary obscat made by READ