The SWIFT XRT Data Reduction Guide

Version 1.2
April 2005

M. Capalbi, M. Perri, B. Saija, F. Tamburelli
(ASI Science Data Center)
&
Lorella Angelini
(HEASARC)

Contents

Chapter 1

INTRODUCTION

1.1 Scope

This Guide describes the principles of the processing and reduction of *Swift* data taken with the X-ray Telescope (XRT) instrument. By reduction, we mean the preparation of data for analysis, a process which entails first calibration and screening of the data and then selecting the desired parts of the screened data from which higher-level data products (i.e., spectra, light curves and images) could be extracted.
This guide assumes that the data have already been downloaded from the archive and that the Swift software and calibration data provided in CALDB are installed and initialized.
The data reduction procedure for the Swift XRT uses tools that account for the calibration, aspect and algorithms specific to the XRT (XRTDAS) as well as generic tools, FTOOLs, used to manipulate the FITS data files. The main focus of this Guide is on :

- *Swift* XRT data files,
- the properties of the *Swift* XRT instrument and its modes, and
- the criteria for identifying good and bad data.

1.2 The Basic Scheme

The XRT *Swift* data are converted into FITS files at the Swift Data Center (SDC) which also runs the XRT pipeline. The pipeline outputs different levels of science data, which are subsequently archived, corresponding to stages of the processing pipeline. It also produces a filter file (`mkf` file), which contains the time-histories of various parameters to which good data can be referenced, identified and screened.
The stages of the pipeline include standard calibration, screening and filtering. At the first stage the science data are calibrated. When screening the data, the appropriate tools consult the science data files and the accompanying `mkf` files to produce a list of selected Good Time Intervals (GTI). These GTIs are used for extracting a list of screened events (maintaining the same FITS structure). The final stage is the filtering (spatial, temporal or spectral) of the events list, which is then binned appropriately for the extraction, of higher level data products in the standard FITS formats. The products, spectra, light curves, and images, can be read into multi–mission data analysis programs such as `XSPEC`, `XRONOS` and `XIMAGE`, respectively, or into any other packages that can handle these formats. Users can reproduce any stage of the pipeline, and therefore any level of the science

data, either because of improved calibration files or because they wish to apply different screening and filtering criteria. This requires the use of a set of Swift XRT-specific and other multi-mission FTOOLS. Since the usage of these tools is repetitive, a script called `xrtpipeline` has been produced to take care of this task. `xrtpipeline` is the usual starting point in the reduction of *Swift* XRT data.

1.3 Organization of this Guide

- The second chapter describes the aspects of the XRT data modes that are related to data reduction and analysis. The special reduction techniques required by the various instrument modes are described.
- The third chapter is devoted to a brief description of the XRT FITS data files and of the `mkf` filter file, since familiarity with the basic structure of these files is important when reducing XRT data.
- The fourth chapter gives a description of the steps involved in the data reduction and the specific XRT tools used.
- The fifth chapter describes the generic screening criteria that must be applied to the data sets before these can be analyzed.
- The sixth chapter covers the next stage of the data reduction, namely how to filter subsets of your screened data before creating data products and the extraction of spectra light curves and images. Includes also example how to use these products.
- The seventh chapter is dedicated to the TDRSS messages.
- The eighth chapter is dedicated to the calibration files used in the data reduction software.
- First appendix : list of table formats for the science files.
- Second appendix : list of most common warnings and errors and possible solution.
- Third appendix : list of the individual helps for each of the XRT specific tasks.

1.4 New releases and Updates

This version of the guide is written based on the Swift software release version 2 that was exercised on data from the performance verification phase. During the performance verification activities, improvements and changes of the software, driven by the Swift observations and the on-orbit calibration, and the failure of the cooling system on the XRT have been incorporated in the current Swift software release and the guide updated accordingly.
The latest information on new software and calibration releases are posted at:

`http://swift.gsfc.nasa.gov/`

Request of additional information and bug reports can be entered in the 'Feedback form' located at that URL.

Chapter 2

XRT modes

2.1 Introduction

This chapter describes the aspects of the XRT performance which users should be aware when reducing and analyzing data. In particular, the various XRT data modes are discussed alongside the special analysis techniques they require. The XRT uses grazing incidence Wolter I mirror (originally built for Jet-X) to focus X-rays onto a CCD detector similar to the EPIC MOS detector flown on XMM. The main XRT characteristics are listed in Table 1 and a complete description of the instrument is given in Burrows et al. 2003 (SPIE, 4851, 1320) and Hill et al. 2004 (SPIE,5165,217).

The dimension of the CCD on the XRT is 600x602 pixels and it is equipped with four calibration sources located at each corner of the detector. The energy of the of the sources are 5.9 keV and 6.4 keV. The location and the radius of the calibration sources in detector coordinates (see later the definition) are:

- Circle (35, 570,47) Cal 0
- Circle (573, 561,48) Cal 1
- Circle (36, 27,47) Cal 2

Table 1: XRT Characteristics

Telescope:	Wolter I (3.5 m focal length)
Detector:	E2V CCD-22
Pixel Size:	40μ m X 40μ m
Pixel Scale :	2.36 arcsec per pixel
Field of View :	23.6 X 23.6 arcmin
PSF:	18 arcsec HPD at 1.5 keV
	22 arcsec HPD at 8.1 keV
Position accuracy :	3 arcsec
Energy Range :	0.2-10 keV
Energy Resolution:	140 eV at 5.9 keV (at launch)
Effective Area:	135 cm^2 at 1.5 keV
	20 cm^2 at 8.1 keV
Sensitivity :	2×10^{-14} erg/cm2/s at 10^4 sec
	in Photon Counting mode

- Circle (576, 20,44) Cal 3

i.e., calibration source 0 is at location (DETX, DETY)=(35,570) and has a radius of 47 pixels.

2.2 Description of XRT Modes

The XRT can operate in two states: Auto and Manual state. The Manual state is used for calibration and the science modes can be commanded for a given observation. In Auto state the XRT automatically select the science mode according to the source count rate. The Auto state is the normal operating mode.

The XRT can operate in the following science modes in either Auto or Manual state:

- Image Long and Short (IM)
- Low rate (LR) and Piled-up Photodiode (PU)
- Windowed Timing (WT)
- Photon Counting (PC)

In Auto state, the sequence in which the modes are scheduled on board and the exposure time depend on the source brightness. Several parameters can be set for each of the modes and have been optimized since the beginning of the mission and they are listed at the end of this chapter. The Photodiode and Windowed Timing modes are also referenced within this guide as "Timing modes". The Swift observatory has two main observation types Automatic Target (AT), when a new GRB is detected and the satellite autonomously slews to the new position and Pre-Planned Target (PPT), when the observation is planned on ground and up-loaded to Swift. When observing a new GRB (AT) the XRT automatically schedules the different modes as shown in Figure 1. First it takes an image in Image mode to calculate the on-board source position and after run in sequence the following modes: Photodiode, Windowed Timing and Photon Counting, switching automatically between modes according with the source intensity. For PPT the same sequence is followed with the exception that the Image mode is not scheduled. When the spacecraft settle on the AT, the first data taken are analysed on-board and the results are sent to the ground via TDRSS as messages and distributed via the GCN. The content of these messages are described in the Chapter 7. Data from an Automatic or Pre-Planned Target observation are transmitted to the ground via the Malindi station.

The following sections describe the characteristics of the individual modes. A short summary is given in table 2.1 together with the flux level at which the XRT switches between modes.

In general, all modes are operated in high gain from the amplifier 1 except for imaging mode which requires a larger full scale response for the brighest burst and therefore is read out from amplifier 2 and in low gain.

2.2.1 Photodiode modes

The Photodiode mode is designed for very bright sources and for high time resolution. This mode performs one serial clock shift and one parallel clock shift alternately and the result is a very rapid clocking of each pixel across any given point on the CCD. The charge is accumulated in the serial register during each parallel transfer, with the result that each pixel contains charge integrated over the entire field of view but not from the same instant in time. The stream of data is telemetered

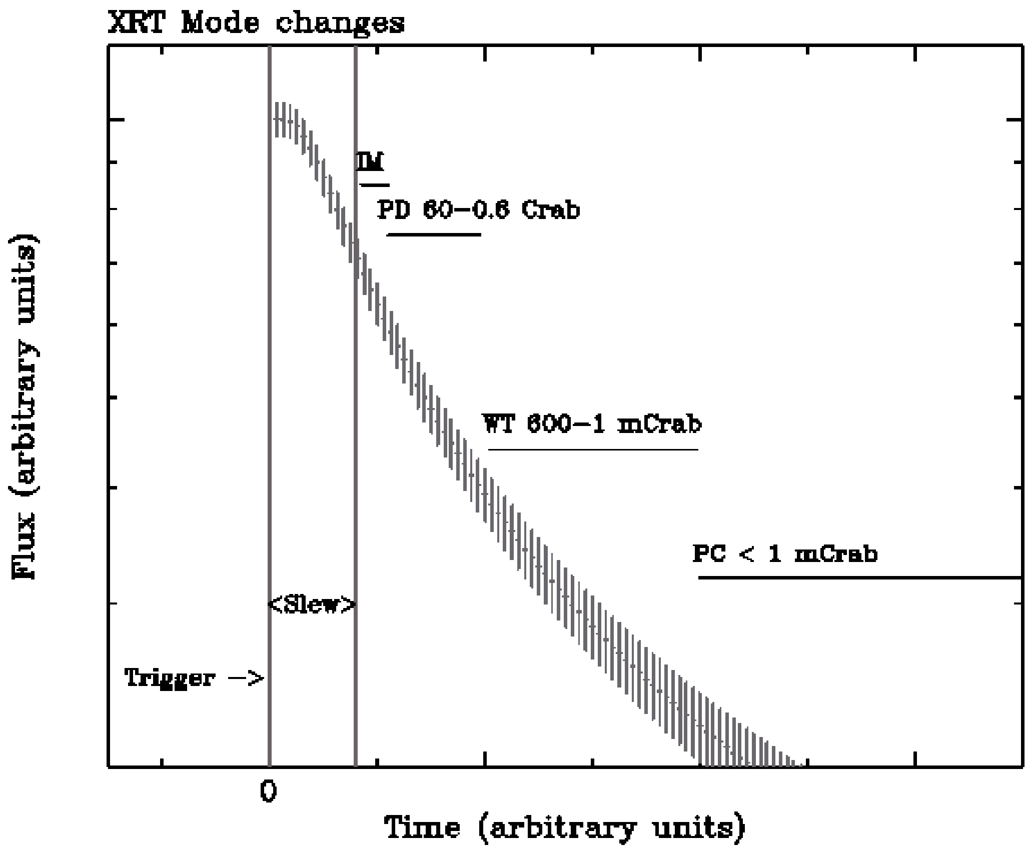

Figure 2.1: Sequence of the XRT mode for an Automatic Target

Mode	Image capability	Spectral Capability	Time resolution	Cal sources in FOV	On-board Event reconstruction	Flux level mode switch
PU & LR	no	Yes	0.14 ms	yes	no, done on-ground	0.6-60 Crab
WT	1D	Yes	1.7 ms	no	no, done on ground	1-600 mCrab
PC	2D	Yes	2.5 s	See window size	yes	< 1 mCrab
IM	2D	No	0.1 s (short)	yes	not applicable	> 140 mCrab
		No	2.5 s (long)			< 5.6 mCrab

Table 2.1: Summary of the XRT mode characteristics. Note: the * indicates that in Piled-up mode the spectral capability is limited because if the source flux is too high the spectrum is piled-up .

in 'pseudo-frames' consisting of events from N rows and 602 pixels, where N is a commandable parameter. This mode is used when the image is dominated by a bright GRB or a very strong source. Photodiode mode thus does not have spatial information but does produce a high resolution lightcurve and a spectrum.

The data are telemetered in two different ways : Low rate and Piled-up. In the Low rate mode only pixels above the lower level discriminator threshold are sent down, whereas in the Piled-up mode all pixels in the 'pseudo-frame' are sent down resulting in a more efficent telemetry format.

The on-board software can be set to either subtract the bias on-board before sending down

the data (default), or to send down the data without bias subtraction. In Piled-up mode the bias calculated in the last Low-rate frame is used for the bias subtraction. In Low-rate the bias is recalculated every frame. The timing information is inserted into every frame of the telemetry stream. The time tag for each pixel is performed on the ground and requires the frame start time and the knowledge of the source location on the CCD.
The algorithm used to time tag the events assumes that the field of view is dominated by a single bright source and the approximation used is that every photon arrives on the CCD at the source position.
The first 1850 pixels in the first frame taken in Photodiode mode are not fully exposed due to the fact that there is a time delay before all pixels in the serial register have constant sky and background exposure. These partially exposed pixels are removed by the ground software.

Typically, in a CCD detector the charge cloud produced by an X-ray photon is not localized into one pixel but it is spread out over several pixels. The event reconstruction is not done on-board and therefore the telemetry does not include for each event the neighborhood matrix. The event reconstruction is performed on the ground, where pixels are evaluated to determine if the charge in the pixel is due to the main X-ray event or its diffusion. This process assigns the grade and the PHA value for each valid event.

In Photodiode mode the signals from calibration sources are mixed with the data and this has to be taken into account during spectral fitting by using an appropriate background spectrum. Since there is no imaging information in this mode, hot and flickering pixels can not be removed, but the impact is minimal due to the clocking speed in this mode. The time resolution of this mode is 0.14 ms. These modes are useful for fluxes up to 60 Crab and typically scheduled at the beginning of observation of a new GRB when the flux is high. Also data during the slews are collected in Low Rate Photodiode.
The initial data processing for the Photodiode mode has to account for the following :

- Remove partially exposed pixels
- Assign the proper arrival time to each event
- Subtract the bias only if the instrument is not configured to subtract the bias or insufficent bias has been sutracted. This is expected to occur sporadically.
- Reconstruct events and assign grade and PHA values. NOTE: During the data reduction of the Low-rate the split threshold can never be set less than the on-board lower level discriminator.

2.2.2 Windowed Timing mode

The Windowed Timing mode is obtained by binning 10 rows in the serial register, i.e. compressing 10 rows into a single row, and then reading out only the central 200 columns of the CCD. It therefore covers the central 8 arcmin of the field of view and one dimensional imaging is preserved. The telemetered information is divided into frames, where each frame contains 600 rows. Similarly to the Photodiode mode, the timing information is inserted every 'pseudo' frame of 600 rows in the telemetry stream and the time tagging of each pixel is performed on the ground. The pixels in the first [60 + 0.5*(600/10)] rows are under exposed and removed during the data reduction. This requires the knowledge of the frame time and of the source position in detector coordinates. The Windowed Timing data are bias-subtracted on-board, and only pixels above the lower level discriminator threshold are telemetered.

Because the window setting includes only the central 200 columns, the calibration sources are not included. Event reconstruction is performed on the ground where grade and PHA values are assigned. Bad columns can be removed corresponding to one image dimension, but due to the fast clocking the charge accumulated from bad and hot pixels is usually below the lower level discriminator. The time resolution of this mode is 1.7 ms. This mode is useful for fluxes between 1-600 mCrab. The initial data processing for the Windowed Timing has to account for the following :

- Remove partially exposed pixels
- Assign the time to each event
- Reconstruct events and assign grade and PHA. NOTE: During the data reduction of the Windowed Timing the split threshold can never be set less than the on-board lower level discriminator.
- Flag bad columns

2.2.3 Photon Counting mode

Photon counting mode retains full imaging and spectroscopic resolution but the time resolution is limited. A full field of view is accumulated every 2.5 sec and the CCD operates in what is known as 'frame-transfer' configuration. Each CCD frame is rapidly transferred into a framestore area, and then read out by clocking the frame store one row at a time into the serial register. The pixels are processed on board where the bias is subtracted, the lower level discriminator is applied and the events are reconstructed. The latter is done by testing if the central pixel of a 3x3 matrix is the local maximum and whether or not it falls between the event discriminator and upper level discriminator thresholds. Then the outer guard ring pixels (a 5x5 matrix) are tested to check if any exceeds the outer ring threshold. This eliminates most of the cosmic rays and chip defects.

For each valid event, the 3x3 matrix is telemetered. On the ground a single PHA value is reconstructed and the grade assigned according to the grade description given in the following sections. The calibration sources are included in the data when the window is set to the full field of view 600x600 pixels and these are removed on the ground when screening the data. During operation for most of the time the standard window setting is smaller (480x480 pixels) excluding the calibration sources and only a few frames for engineering purposes are taken with the full field of view each day. The time resolution of this mode is 2.5 seconds. This mode is useful for fluxes below 1 mCrab and is piled-up if there is more than 2 source count per frame.
The initial data processing for the Photon Counting mode has to do for the following :

- Flag bad pixels (as defined in CALDB)
- Flag thresholded events
- Flag calibration sources
- Calculate and Flag hot and flickering pixels
- Assign grade and PHA values

2.2.4 Image mode

Image mode is used by the XRT to obtain a rapid position of a new GRB. If the spacecraft slews to a new GRB, the XRT takes an image and processes the image on-board to determine its position. The CCD operates like an optical CCD, collecting the accumulated charge on the detector and reading out without any X-ray event recognition. The image will be highly piled-up and produces no spectroscopy data, but it is used to derive accurate position and flux estimates. The Image mode operates in low gain and can be used for fluxes between 25 mCrab and 45 Crab. Due to how the Image mode operate, the image is not a 2D histogram of the number of events but each pixel contains a DN (DN= Data Number, the native units for the amplifier's analog-to-digital converter) value proportional to the total charge accumulated in that pixel during an exposure.

Only pixels exceeding the lower level discriminator threshold are sent down. The detector bias is not subtracted on-board and also the calibration sources maybe included depending on the window setting. Depending on the source flux the exposure of image mode is automatically set on-board either to 0.1 or 2.5 seconds.
The initial data processing for the Imaging mode has to do the following:

- Subtract the bias
- Clean calibration sources
- Clean bad pixels

2.3 Classification of Events and Grade

To eliminate events due to charged particles and to obtain the expected energy resolution, X-ray events from each readout are identified and classified. For the Photon Counting mode the distribution of the charge in the 3x3 matrix is classified according to a library of 32 grades (See fig. 2.2). For the Photon counting mode, grades in the range of 0-12 are considered good grades.

For Windowed Timing and Photodiode modes, it is not possible to use the above grade definition since the 3x3 matrix information is not available. A 7x1 matrix is instead used to reconstruct the events and to grade them according to a library of 15 grades (See fig. 2.3).

For Windowed Timing mode, grades in the range of 0-2 are considered good, instead the range between 0-5 are considered good for the Photodiode mode.

2.4 XRT configuration: Changes post-launch

Since launch the following configuration settings and parameters have been changed to optimize the instrument performance :

- Pre-Camera Door opens (Launch-12 Dec 2004): During the activation of the XRT it was discovered a problem with the Thermo-Electic Cooler that prevents to run the instrument at -100 Celsius. Running the instrument at highertemperature led to a number of configuration changes starting soon after the door was open throughout the performance verification phase. Other activities pre-opening the door included calibration of the gain and bias with door sources and testing various upper and lower level discriminator for all modes as function of temperature.

- 12 Dec 2004 Camera Door open : First light on Cas A.
- 13-15 Dec 2004 Alignment to star tracker using optically bright stars. Update of the star tracker parameters on 1:55 UTC on 15th Dec and verify till the 18th of December. Data before the Dec 15th show a $\sim 3arcmin$ offset.
- 13 Dec 2004-21 Jan 2005: Period in which instrument and spacecraft operation were optimized to control the XRT temperature Bias varies strongly with temperature and the bias thresholds were updated to mitigate the problem. Data taken during this period are to be analysed always by looking at the temperature first and discard any data when the CCD temperature is higher than -50 Celsius.
- 18 Dec 2005 Update the TAM reference positions. After this date the on-board position are correctly adjuted for the TAM correction.
- 30 Dec 2004 Change the default PC window setting from 490x490 to 480x480. The XRT FOV completly excludes the corner sources in the Photon counting mode.
- 18-21 Jan-2005 Calibration observations to verify the on-board position accuracy.
- 31 Jan 2005 (at 21:57 UTC) Remove the velocity aiding. The data taken before the 31 of January should be analyzed using the corrected attitude now available with data that were processed after Feb 2005.
- 1 Mar 2005 13:23 Decrease the size of the 'pseudo-frame' for the LR and PU modes, generating more frames per unit time
- 1 Mar 2005 13:23 Change the count rate level switch point from PC to WT for the Auto state. The switch point was incresed from 5 count/sec to 10 count/sec. This may cause pile-up if the source re-brightens when already the XRT has switched to PC.
- 14 Mar 2005 at 15:22 UTC Change the lower level discriminator to 70 for WT and PC modes. This change eliminates most of the bright earth effect which cause the bias to be underestimated.
- 14 Mar 2005 15:22 Upload final setting for the Image mode data to limit the number of false centroids calculated on-board in the case of faint GRB.
- 22 Mar 2005 12:30 Change the lower level discriminator to 80 for LR, PU , WT and PC modes. Better threshold esimated from ground analysis and uploaded on the spacecraft.

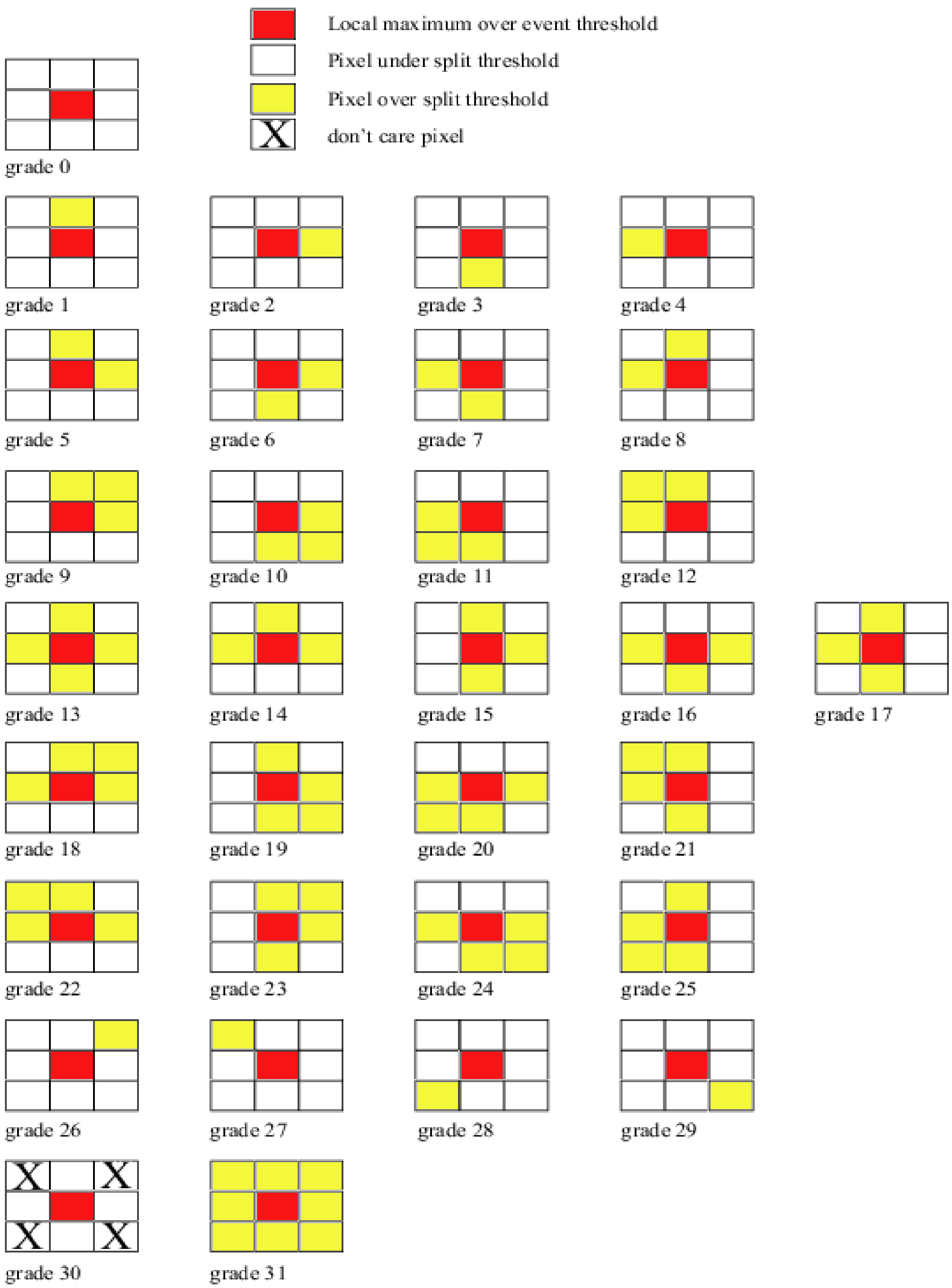

Figure 2.2: The definition of the XRT grades for Photon Counting mode

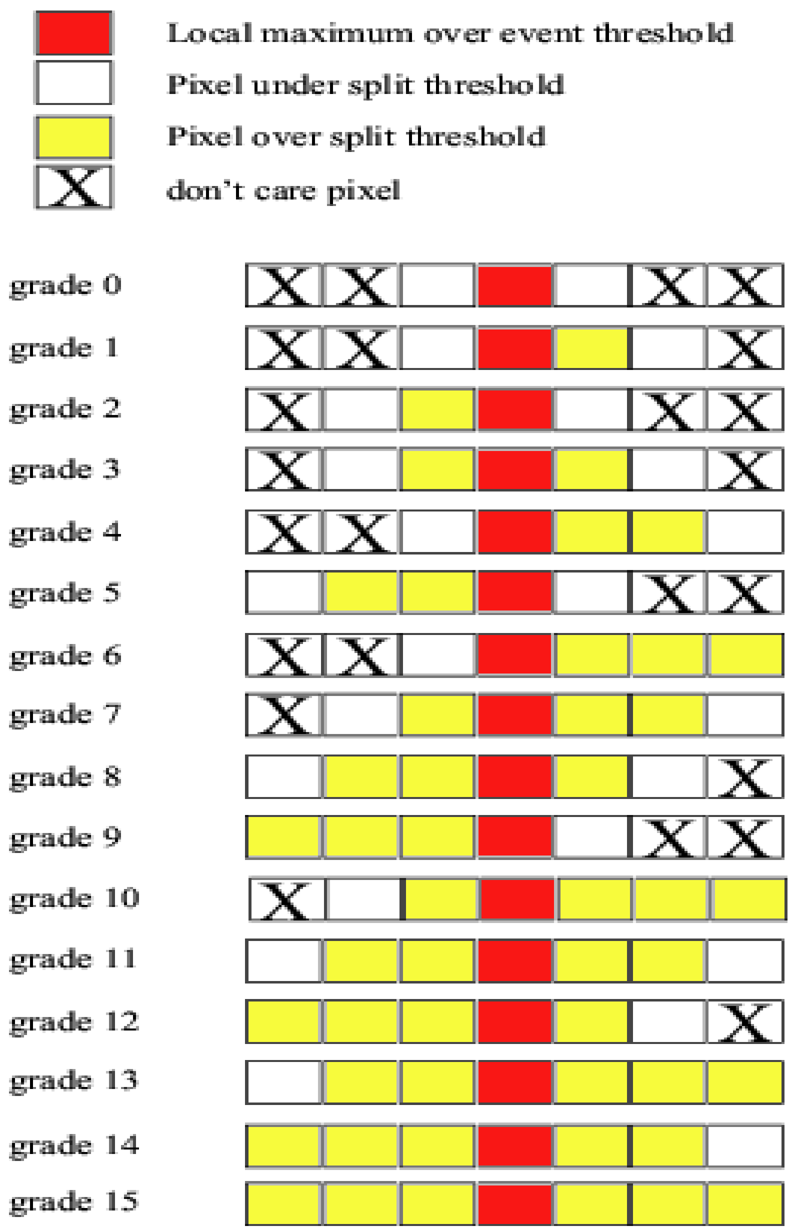

Figure 2.3: The definition of the XRT grades for Timing modes

Chapter 3

DATA FILES

3.1 Introduction

The *Swift* XRT science data can be collected using different readout modes, as discussed in the chapter 2. These are the Image, Piled Up Photodiode, Low Rate Photodiode, Windowed Timing, and Photon Counting modes. This chapter includes the description of the XRT science data files included in an observation, either from an Automatic or Pre-Planned Target. The content of the TDRSS data is described in 7. An observation includes several orbits, that are not contiguous. The data are distributed in FITS format files, each dedicated to a specific mode.

With the exception of Image mode, the structure of all Science files is an event list where each row includes parameters and associated flags for each event. For the Imaging mode, the data are stored using the FITS image extension, i.e. an image array. The FITS layout and keywords of the files follow the OGIP standards.

3.2 Basic file structure, Levels of *Swift* XRT Data and Filename

3.2.1 Events FITS File Structure

All the XRT *Swift* data in FITS event format have the following file structure :

1. Primary header,
2. Events extension,
3. Good Time Interval extension,
4. Bad pixel table.

The primary header contains general information about the mission, the instrument and the observation identifier. As with all FITS headers, this information is in the form of keywords with assigned values. No data are included. The event extension is in the form of a binary table called EVENTS, and contains complete description of the events themselves in the form of a time-ordered list of photon attributes, e.g. time, position and pulse-height information. The listed attributes depend on the mode. The GTI extension contains time intervals of good data according to the level of the processing. The GTI binary table has the same structure for all modes and for all levels
The ground software adds a BADPIX extension in the event files for the Photon Counting and Windowed Timing modes containing the position of the pixels or columns that are flagged bad.

3.2.2 Image FITS File Structure

Data taken with the Image mode are stored in FITS image extension with the following file structure:

1. Primary header,
2. Image extension (s)

The primary header is similar to that of the event files. This is followed by image extensions as many as the number of exposure taken with the Image mode. Typically only one exposure is taken during normal operation. Therefore each image extension correspond a single exposure. Because of the Image mode characteristics, each pixel of the image array stores the intergrated charge collected in that pixel and does not correlate to a single count.

3.2.3 The Levels of *Swift* XRT Data

There are two different levels of files produced by the processing and archived. These are:

1. **Level 1 and Level 1a.** The Level 1 event and image files are produced by a task that reformats the telemetry into FITS files. No information is lost in this process and additional information is calculated and added to the file. For each observation there is one file per readout mode. However if the window size is changed on-board (for the Windowed Timing and Photon Counting modes) or if the bias subtraction is switched on/off (in Photodiode modes) there is a different event file for each configuration. The data are taken in Low-rate Photodiode mode during the slews and the settling are stored in a separate file. For the Photodiode and Windowed Timing modes there is an intermediate level (1a) of files which is needed to associate the proper arrival time with the events (the telemetry reports just the frame time) and to reconstruct the events to assign grade and PHA values as this is not done on board for the timing modes.
   Users that wish to apply their own screening should start from the Level 1 for the Photon Counting mode and Level 1a for the timing modes while users that wish to use different software default settings or another source position should start from the Level 1 for all modes.

2. **Level 2.** The Level 2 event files have been calibrated and screened through a standard screening process (see chapter 5 on standard screening). The structure is the same as Level 1 file, but depending on the mode, some of the columns may have been removed. Users can read the Level 2 event files into `XSELECT` and proceed immediately to extract higher level products such as images, light curves and spectra. The Level 2 image files from the Image mode have been calibrated and the sky coordinates included in the header aligned with the celestial north.

3.2.4 XRT file naming convention

The file name format for the Swift XRT science files uses the following convention:

Event : 'sw[obs_id]x[mm]*[ww][pp]*_[lev].[ext]'

and

Image : ’sw[obs_id]x[mm]_[lev].[ext]’

where:

- **sw** is a prefix to indicate the mission name (Swift).
- **[obs_id]x** The obs_id contains an 11 digits number to identify the observation and the x identifies the instrument (XRT).
- **[mm]** is a two character string that identifies the instrument operating modes. These are listed in the Table 3.1.
- **[ww]** identifies the window setting of the CCD in the Photon Counting and Windowed Timing event files. For the Photodiode modes instead it is used to identify if the bias has been subtracted on-board or not. In Photon Counting and Windowed Timing modes, this is set to ’wN’ where is a running number from 1-9 and corresponds to a specific CCD window setting in each of the modes. The pre-launch settings are listed in the Table (3.1). In the Photodiode mode, this is set to ’bN’ and identifies if the bias has been subtracted on-board or not, where N is either 0 (not subtracted) or 1. c=swifthea
- **[pp]** identifies if the event data were taken with the satellite in pointing mode *po*, or during a slew *sl* or during a settling phase *sd* (within 10 arcmin of the targets).
- **[lev]** gives the file level. The level is not a number, but an additional specifier to distinguish between different stages of processing; This is set to *uf* for the Level 1, *ufre* for the Level 1a and *cl* for the Level 2 event files. In the image file this is set to *rw* for the Level 1 and *sk* for the Level 2
- **[ext]** the file extension. This is set to *.evt* for event files and to *.img* for the image files

Mode	mm
Photon Counting	pc
Windowed	wt
Piled Up	pu
Low Rate	lr
Imaging	im

ww	Windowed Timing	Photon counting
w1	100 columns	490x490 pixel
w2	200 columns	500x500 pixel
w3	300 columns	600x600 pixel
w4	400 columns	480X480 pixel
w5	500 columns	

ww	Photodiode
b0	Bias not subtracted
b1	Bias subtracted

Table 3.1: Values for sub-mode mm and ww

3.3 Main columns in *Swift* XRT FITS Events Files

This section describes the important columns found in the event files. The complete listing of the columns for each of the event files is provided in the appendix A.

3.3.1 RAWX/Y, DETX/Y, X/Y and OFFSET Columns

For the XRT Photon Counting mode, the RAWX and RAWY columns give the discrete CCD pixel location of each event processed by the on-board electronics. The data processing, using the ground and in-flight calibrations, produces the DETX and DETY focal plane coordinates and X and Y sky coordinates. The data in each pair of columns are in units of pixels (0.040 mm/pixel; 1 pixel = 2.36 arcsec). In the RAW coordinates, the pixels are numbered relative to the output amplifier. The DET cooordinates are the focal plane coordinates. They are amplifier independent, in a 'looking down ' orientation and with the DETY flipped relative to the spacecraft coordinate system. The aspect solution is applied to the focal plane coordinates (i.e., DETX, DETY) to produce the sky coordinates (i.e., X, Y). Each event is projected back onto the sky on a tangent plane, and a binned sky image is formed with the X and Y axes oriented along RA and DEC, respectively.

In the Windowed Timing mode, the column RAWX contains the telemetered spatial information, while RAWY is a counter incremented by one when a row (sum of 10 CCD rows) is read out. The Photodiode mode does not have positional information at all, while the temporal information is put into the OFFSET column. To reconstruct the photon arrival time for a given pixel in Timing modes, it is necessary to know the location on the CCD of the source image. Ground software assumes that the CCD is dominated by at most one bright source, so it is possible to make the approximation that every photon arrived at the source position. In Windowed Timing mode, the DETX is the focal plane coordinates transformed starting from the RAWX coordinate, while the DETY corresponds to the assumed source position used to calculate the photon arrival time of the events. The X and Y columns are the sky coordinates where the Y values are obtained from the source position. In Photodiode mode, DETX and DETY are both corresponding to the source position used to time tag the events. If data are taken during the slew, the software uses the detector coordinates of the center of the CCD and filled these columns accordingly. Photodiode mode data do not contain X and Y columns.

3.3.2 TIME and ROTIME Columns

The TIME column contains the time assigned to each event and it is given in seconds after the reference time. This is January 1st, 2001 UTC. In the header of the FITS file the reference time is expressed in the TT time system and the values (in units of days) are written in the keywords MJDREFI and MJDREFF. The keywords values are the following :

```
TIMESYS=  'TT'
MJDREFI = 51910
MJDREFF = 7.4287037e-4
```

where MJDREFF contains the offset between the UTC and TT on January 1 2001. The time value stored in the TIME column has different meaning depending on the mode:

- Photon Counting mode:
  The values in the TIME column are the CCD exposure start times This means that all the events within the same readout have the same time. The time resolution of these data is 2.5073 seconds.

- Windowed Timing and Photodiode modes :
  The values in the TIME column corresponding to the arrival time of the photons are derived by the ground software using the source position. The timing information telemetered that give the time associated to each read-out frame is stored in the ROTIME column.

The times in the columns always reference to the beginning of the integration time. This is recorded in the header of the fits file in the keyword TIMEPIXR set to 0.

3.3.3 PHA, PHAS, GRADE and PI Columns

The existence and the content of these columns depend on the data mode and on the file level.

- PHAS:
  For Photon Counting, the telemetry contains the DN values of the event and a 3x3 pixels neighborhood centered on this. These values are stored in the PHAS vector column of the Level 1 FITS file as a 9 element array. The first element PHAS[1] corresponds to the central pixel of the 3x3 array; PHAS[2] and PHAS[4] are the lower left and right corner pixels and the PHAS[7] and PHAS[9] are the upper left and right corner pixels of the 3x3 array.
  For Timing modes, the event recognition is done by the ground software, using a 7x1 vector. Therefore the Level 1 of the timing modes does not contain the PHAS column, but optionally this is added by the processing in the Level 1a.

- PHA and GRADE:
  The ground software calculates a single PHA value for each event and a number that describes the grade of the event stored the GRADE column. For Photon Counting mode these columns are filled in the Level 1 FITS file. For Timing modes, the PHA column of the Level 1 file contains the DN values telemetered, while in the Level 1a the PHA column contains values calculated by ground software after event reconstruction and the content of the PHA column is copied into a new column named EVTPHA.

- PI:
  the PI (Pulse Invariant) column is derived by gain-correcting the PHA values. This column is filled by ground software in Level 1 files for Photon Counting and Level 1a for Timing Modes.

3.3.4 STATUS Column

This column contains a bit mask flag describing the quality of the event. The column is populated during the data calibration that creates the Level 1 or 1a files.

The possible flags are :

b0000000000000000 Good event

b0000000000000001 Event falls in bad pixel from CALDB

b0000000000000010 Event falls in bad pixel from on board Bad Pixels Table

b0000000000000100 Event falls in dead pixel

b0000000000001000 Event falls in hot pixel

b0000000000010000 Event falls in user bad pixel

b0000000000100000 Point

b0000000001000000 Column

b0000000010000000 Event has PHAS[1]< Event Threshold

b0000000100000000 Event has a neighbor bad from bad pixels list

b0000001000000000 Bad event

b0000010000000000 Event from calibration source 1

b0000100000000000 Event from calibration source 2

b0001000000000000 Event from calibration source 3

b0010000000000000 Event from calibration source 4

b0100000000000000 Saturated pixel

b1000000000000000 Event falls in flickering bad pixel

3.3.5 Other Relevant Columns

There are also the following additional columns

- Amp
  contains the value of the amplifier used to read the data.
- CCDFrame
  contains the frame number. Note the values within this column are recycled for each orbit during an observation.
- PixsAbove
  contains the number of pixels above the split threshold which are considered in the computation of the PHA value. It is an optional column for Timing modes, which the users can add to the Level 1a FITS file.
- RAWPHA
  is present only in Photodiode FITS files and contains the original telemetered DN value before the bias subtraction if it is performed.
- EVTPHA
  is added by the ground software to the Level 1a FITS files, before event reconstruction moves the content of the PHA column into it.

3.4 Other relevant *Swift* XRT Data files

3.4.1 Housekeeping Files

Within an observation, the XRT has three types of housekeeping file. Two contain the information stored in the header and trailer of the science packet data the last contains engineering values. The HK file structure consists of an empty primary header and a FITS binary extension. The file name uses the following convention :

Housekeeping : 'sw[obs_id]x[hh].hk'

where [hh] is set to *hd* and *tr* to identify the hk file containing the values stored in the header and trailer of the science data packet respectively, and *en* for the engineering data.

The *hd* file is used in the XRT data reduction software. The *hd* file includes columns for each of the parameters found in the header of the science packet regardless if the parameter is common to all the read-out modes. Each row contains the values found in one frame of a specific mode.

The appendix A lists all the columns present in the *hd* file. The XRT ground software uses the following columns contained in this file to build the filter file (See section A.7 for details):

- **CCDTemp**:
  the CCD temperature given in Celsius;
- **PixGtULD**:
  the total number of pixels with DN greater than the Upper Level Discriminator;
- **Vod1** and **Vod2**:
  the measured values of the Output Drain Voltage for Amp1 (left Amp) and Amp2 (right Amp);
- **Vrd1** and **Vrd2**:
  the Reference Voltage for Amp1 and Amp2;
- **Vsub**:
  the measured Substrate Bias Voltage;
- **Vbackjun**:
  the Back Junction Bias Voltage;
- **Baselin1** and **Baselin2**:
  Baseline Voltage for Signal Chain A and Chain B.
- **TIME**:
  CCD frame Start Time.
- **ENDTIME**:
  CCD frame End Time.

The chapter 5 describes how these parameters are used and the ranges allowed.

3.4.2 Filter File

The Filter file or `mkf` file contains a subset of housekeeping, orbit and attitude information, as well as derived parameters such as Earth elevation angle which are useful for the XRT data screening. The filter file is generated on ground by the task `makefilter`. These information are stored in a binary table extension of the `mkf` file in the form of a time-ordered list of rows. When given a screening criterion, e.g. elevation angle > 10 degrees, the selection tool will go through the `mkf` file to find all the Good Time Intervals (GTI) which satisfy the specified criterion. The column included in the `mkf` file used by the XRT software are listed in the appendix A.

Chapter 4

Data Reduction

4.1 Introduction

In this chapter we describe the steps that are involved in creating the screened event files. The starting point is the Level 1 data as stored in the archive, and we review which tasks and the order in which they should be applied during the processing. The same steps described here are coded in the perl script `'xrtpipeline'`.

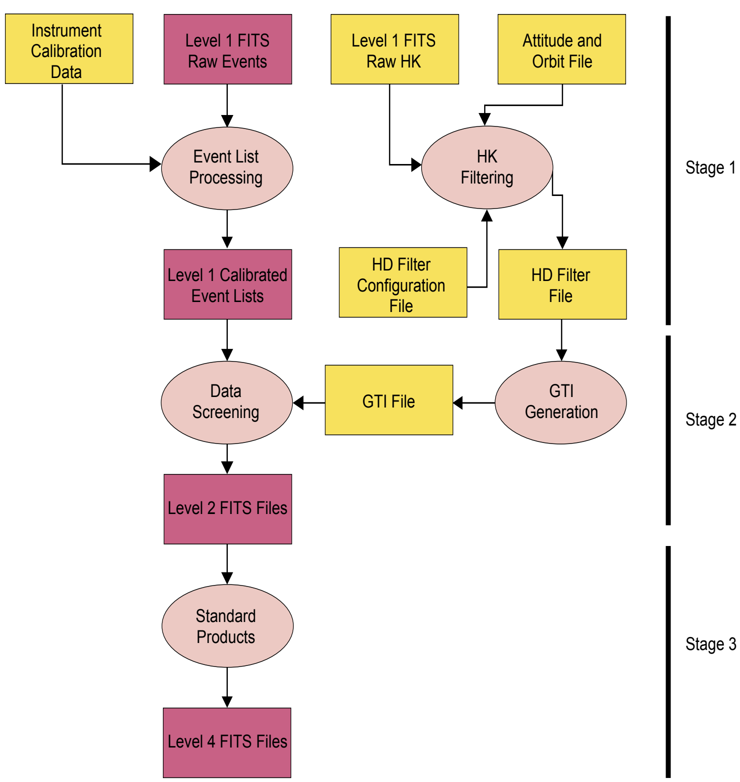

Figure 4.1: The flow diagram of the XRTDAS pipeline

To generate cleaned event files, there are two main stages:

- Stage 1. At this stage the data are calibrated. There are many differences in the procedure to calibrate the Photon Counting and the Timing modes (Windowed Timing and Photodiode), mainly because of the different information telemetered for these modes. The processing involves identification of bad pixels or bad columns, coordinates transformation, time tagging of events, reconstruction of events, computation of the PHA and PI values, and elimination of the piledup frames and partially exposed pixels.
- Stage 2. This stage mainly screens the events calibrated during the Stage 1 process, by applying conditions for specified parameters. The screening uses the GTIs obtained by setting conditions on instrument-specific housekeeping parameters, and on the attitude and orbit related quantities. Additional selections are applied on the GRADE and the STATUS columns.

The Level 1 (Photon Counting and Timing modes) and Level 1a (Timing modes) data, that are stored in the Swift archive, are the output of the Stage 1 processing. However the same tasks can be re-run if new information, such as improved calibration, attitude or source position, becomes available for these files. The Level 2 files stored in the Swift archive are the output of the Stage 2 process. Users starting from the Stage 1 output files can create customized Level 2 files by applying different screening criteria compared to the criteria applied by the standard processing. The following sections give a walk through of data reduction software to generate the Level 1 and 2 files. The individual software tasks are invoked in the examples with the minimum number of parameters. The individual help of the tasks and the allowed parameters are included in the appendix B of this guide. Also the common software errors and warnings are listed in the from appendix C. The examples for the PC, PU and WT modes use a source position of 171.101 and 14.23 deg in RA and Dec and the XRT Level 1 science and HK data, the attitude and the mkf file from the sequence, '00055250019'. For the Image mode the sequence '00113120000' is used instead. The calibration files used in the XRT tasks are not shown as input parameters in the examples because they are defined in hidden parameters and are automatically taken from CALDB, with the exception of the teldef file used in the general task `'coordinator'`.
The same steps and more are coded in a script `'xrtpipeline'`. Example of how to run `'xrtpipeline'` is included at the end of this chapter.

4.2 Stage 1

4.2.1 Photon Counting mode

To generate the calibrated Photon Counting mode outputs the following steps are run in sequence (see flow chart in Fig 4.2):

- *coordinator*: Transform the coordinates from the raw values telemetered to detector and sky coordinates taking into account the satellite attitude.. The command to run 'coordinator' uses the following inputs parameters:

```
coordinator teldef=swx20010101v004.teldef attfile=sw00055250019sat.fits.gz
            eventfile=sw00055250019xpcw4po_uf.evt ra=171.101 dec=14.23
            randomize=yes aberration=no
```

This task requires the event file, the attitude file, the telescope definition file (teldef) and the source coordinates as inputs.
The standard attitude file can be modified by the measurements of the Telescope Alignment Monitoring (TAM) on board *Swift* and input in place of the standard attitude in 'coordinator'. This corrected attitude allows the position determination accuracy to be refined. By default the standard processing uses the standard attitude non corrected for the TAM.

- *xrtflagpix*: Flags known detector bad pixels and the calibration sources. By default the bad pixel list is read from the relevant CALDB file and the flags are written (or over-written) in the STATUS column of the input file (when 'outfile' is set to NONE). Events corresponding to bad pixels and bad column locations and also events associated with the corner calibration sources are flagged and their values set in the STATUS column, in accordance with the definitions given in section 3.3.4 of the event file. The basic command to run `xrtflagpix` has the following input parameters:

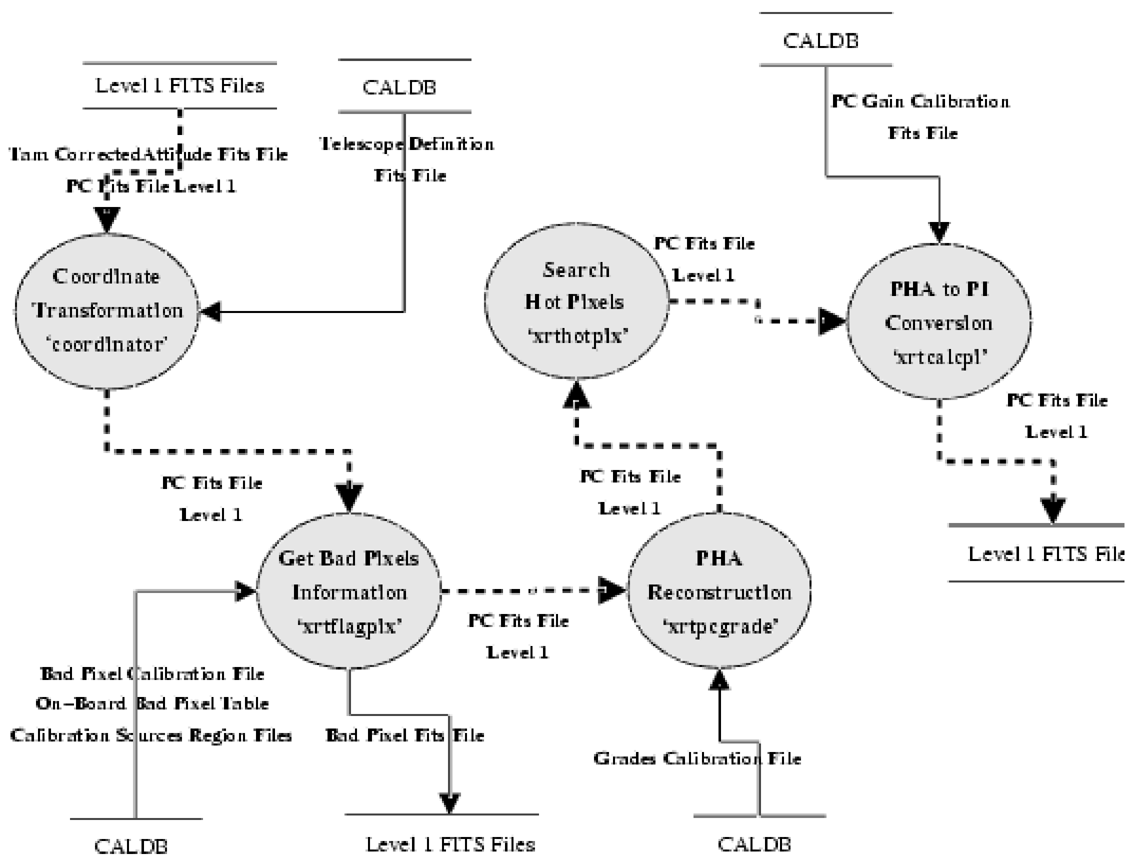

Figure 4.2: The flow diagram for the Stage 1 processing of the Photon Counting modes

```
xrtflagpix outfile=NONE infile=sw00055250019xpcw4po_uf.evt
          hdfile=sw00055250019xhd.hk.gz
```

- *xrtpcgrade*: Calculate a single PHA value and assign a grade to the event. In the Photon Counting mode, each event has associated nine PHA values which are stored in the column

PHAS, corresponding to the central pixel and to the eight surrounding pixels. The basic command to run `xrtpcgrade` has the following input parameters:

```
xrtpcgrade  outfile=NONE infile=sw00055250019xpcw4po_uf.evt
```

The single PHA is calculated by summing all the values in the PHAS column of the pixels above the split threshold. The number of pixels above the split threshold (as defined on board), considered in the computation of the PHA value, is stored in the column named PixsAbove. The split threshold can be changed setting the parameter 'split'. The grade values are assigned according to the scheme given in Chapter 2. The assigned values are stored in the column GRADE.

- *xrthotpix*: Flag anomalous pixels (hot and flickering). Search for hot and flickering pixels by applying a statistical test. The basic command to run `xrthotpix` has the following input parameters:

```
xrthotpix outfile=NONE infile=sw00055250019xpcw4po_uf.evt phamin=0 phamax=4095
```

The pixels with PHA ¡ 'phamin' and PHA ¿ 'phamax' will be ignored by the task.

- *xrtcalcpi* Compute the Pulse Invariant (PI) values. The PI values are calculated accounting for temporal changes in gain, induced by radiation damage, for the gain temperature dependence and for small differences in gain with the position, due to the Charge Transfer Inefficiency (CTI). The basic command is :

```
xrtcalcpi outfile=NONE infile=sw00055250019xpcw4po_uf.evt
          hdfile=sw00055250019xhd.hk.gz
```

The gain is read from the relevant CALDB file and the calculated value is written in the column PI of the input file (when 'outfile' is set to NONE).

4.2.2 Photodiode and Windowed Timing modes

The steps described below are used to calibrate the Photodiode and Windowed Timing modes. The steps for these modes are similar although there are differences due to the difference information telemetered. For each of the steps below it is indicated if it refers to Windowed Timing only or Photodiode only or both.

- *xrthkproc* Compute the frame start and end time using the source coordinates. As explained in Chapter 2 for the Timing modes, the photon arrival time is assigned on ground because it depends on the source position. In order to generate Good Time Intervals (GTI) from the HK parameters, the same algorithm to time tag the events is also applied to the frame times stored in the HK files. The basic command uses the following inputs:

```
xrthkproc hdfile=sw00055250019xhd.hk outfile=sw00055250019xhdtc.hk
          attfile=sw00055250019sat.fits.gz srcdetx=300 srcdety=300
```

where the input is the *hd* HK file found in the archive and srcdetx and srcdety are the position in detector coordinates of the source. This operation is required for both Photodiode and Windowed Timing modes. It should be run for any improved source position and before the task that time tags the events (`xrttimetag`).

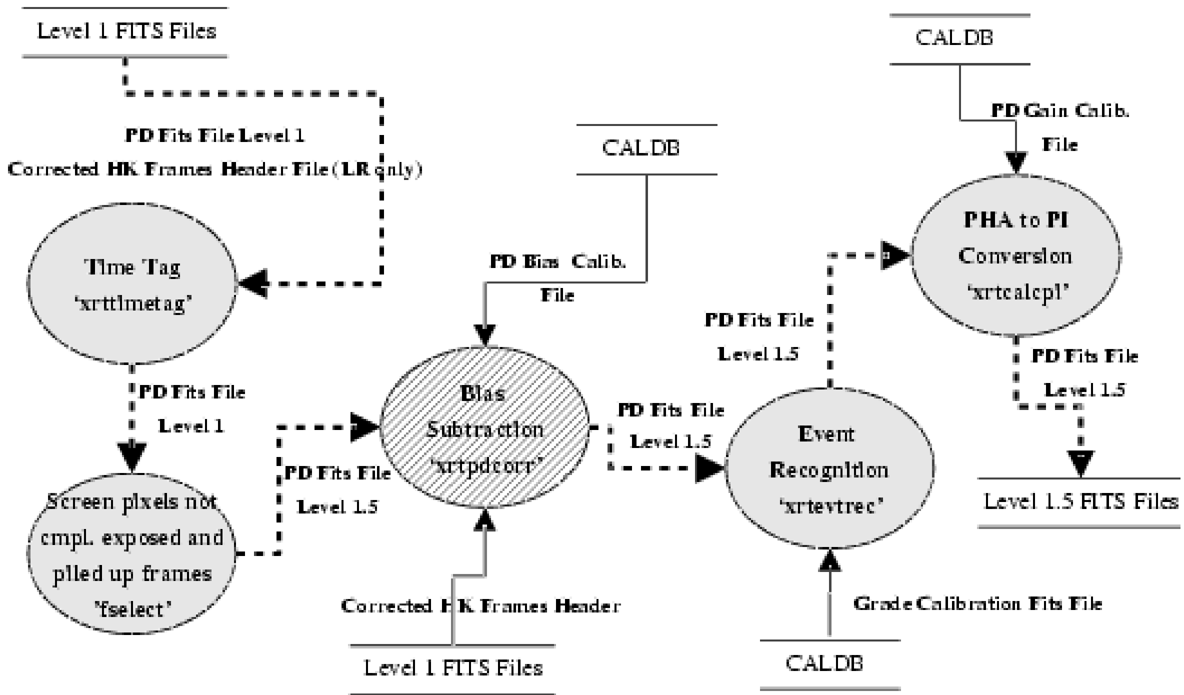

Figure 4.3: The flow diagram for the Stage 1 processing of the Photodiode mode

- *xrtflagpix* Flag bad columns for Windowed Timing mode. Because the Windowed Timing retains only one-dimensional position information, bad columns rather than pixels need to be flagged and later removed from the events list. Note that for Windowed Timing, it is not necessary to flag the corner sources because the calibration sources are already excluded on board from the data. The basic command to run `xrtflagpix` has the following input parameters:

```
xrtflagpix outfile=NONE infile=sw00055250019xwtw2po_uf.evt
```

  By default the bad column list is read from the relevant CALDB file and the flags are written (or over-written) in the STATUS column of the input file (when 'outfile' is set to NONE)

- *xrttimetag* Assign photon arrival times to the events. For the timing modes, the assignment of the photon arrival times is done on the ground since it requires the detector coordinates of the source position. This task also computes the appropriate event coordinates, the event time and the START and STOP values for the GTI extension of the event file. The basic command for `xrttimetag` uses the following inputs:

```
xrttimetag infile=sw00055250019xlrb1po_uf.evt outfile=NONE
           hdfile=sw00055250019xhdtc.hk attfile=sw00055250019sat.fits.gz
           usehkkey=no usesrcdet=no srcra=171.101 srcdec=14.23
           ranom=171.101 decnom=14.23 npixels=1850 percent=30
```

  The input parameters 'npixels' and 'percent' are used only for the Photodiode mode, and indicate the number of pixels not fully exposed and the percentage of events over the ULD. To exclude the not fully exposed events a GTI extension is calculated, however these event are not filtered out by this task. The input parameter 'usesrcdet' flags if the task uses the

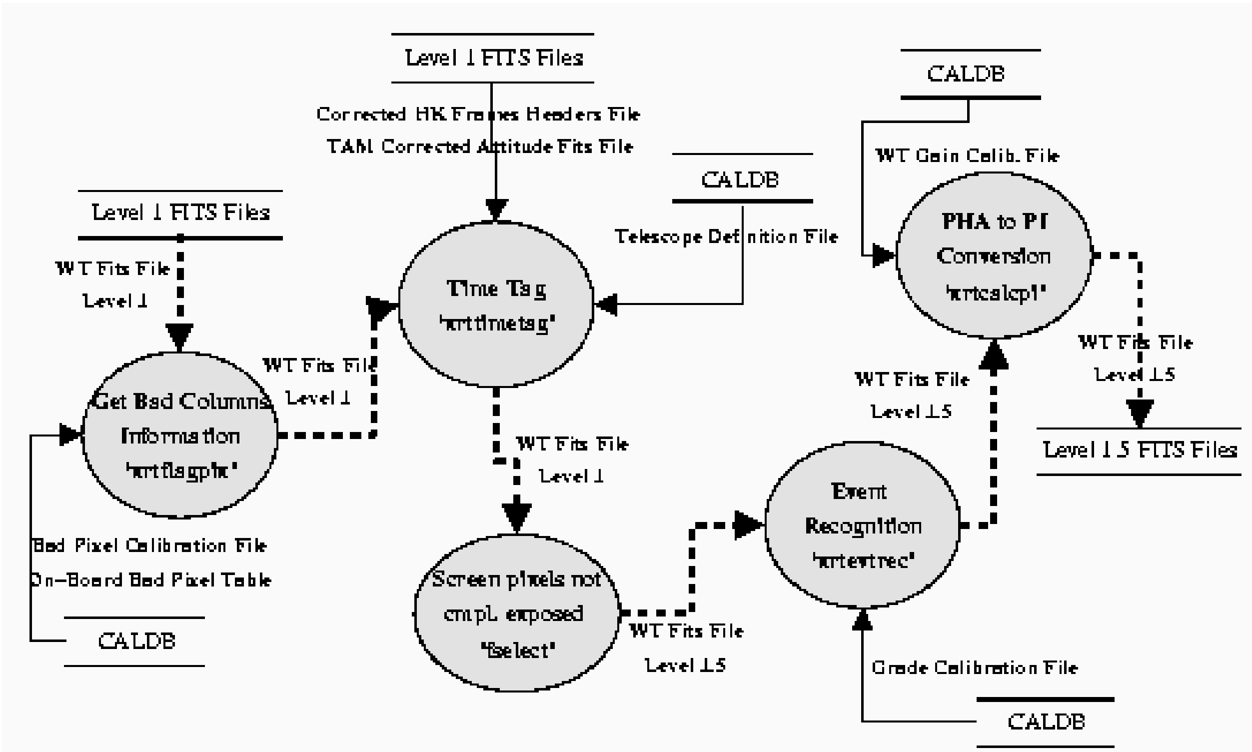

Figure 4.4: The flow diagram for the Stage 1 processing of the Windowed Timing mode

source ra and dec (as in the above example) or the detector coordinates via the parameters 'srcdetx' and 'srcdety'.

- *fselect* Remove pixels not fully exposed for the timing modes. While 'xrttimetag' calculates GTIs to exclude pixels that are not fully exposed, these events are still present in the file. Using 'fselect' with the following input parameters applies the GTI and removes the not fully exposed events from the file :

```
fselect outfile=sw00055250019xlrb1po_ufre.evt expr=gtifilter\(\)
        infile=sw00055250019xlrb1po_uf.evt
```

  These pixels have to be removed before going further with data analysis to avoid including the non fully exposed pixels in the event recognition calculation and the grade assignment.

- *xrtpdcorr* Remove the bias from the Photodiode mode. PHA values for the Photodiode mode are by default telemetered with the bias already subtracted. However the on-board software can be set to sent data without subtracting the bias. If this occurs, the bias subtraction is done on ground using the task `xrtpdcorr` with the command:

```
xrtpdcorr infile=sw00055952001xlrb0po_uf.evt hdfile=sw00055952001xhd.hk.gz
          outfile=NONE  method=SG
```

  The event file has an header keyword BIASONBD set to 'T' to flag if the bias has been subtracted on board. `xrtpdcorr` checks this keyword and if set to 'F' calculates and subtracts the bias.

- *xrtevtrec* Reconstruct the events, assign PHA value and GRADE. For Timing modes the event recognition is done on the ground, and is performed by checking the neighborhood of each local maximum which is 7x1 pixel wide. The PHA column is filled with the value obtained by summing up the PHA of the central pixel with that of the surrounding pixels above the split threshold, and an event GRADE is assigned to each pixel that is recognized as valid event. The basic command uses the following input parameters:

  ```
  xrtevtrec infile=sw00055250019xlrb1po_uf.evt event=80 split=80
            outfile=sw00055250019xlrb1po_ufre.evt  hdfile=sw00055250019xhd.hk
  ```

  It requires as input the event and split threshold, which by default are set to the lower level discriminator value.

- *xrtcalcpi* Assign Pulse Invariant (PI) values. As in the Photon Counting mode, the last step for the Timing modes is to compute the PI values. This is done with the same task `xrtcalcpi` used for the Photon Counting mode. The basic input parameters are:

  ```
  xrtcalcpi infile=sw00055250019xlrb1po_ufre.evt outfile=none
            hdfile=sw00055250019xhd.hk
  ```

  where the input files are the outputs of `xrtevtrec`.

4.3 Create a filter file: all modes

- *xrtfilter* Create a filter file from housekeeping and attitude data. The filter file contains a list of parameters, some related to the specific setting of the instrument HK and some due to the attitude data (see Appendix A).
  The basic command to run `xrtfilter` has the following input parameters:

  ```
  xrtfilter ranom=171.101 decnom=14.23 hdfile=sw00055250019xhd.hk outdir=./
            attfile=sw00055250019sat.fits.gz alignfile=swalign20041115v012.fits
  ```

  where the 'infile' is the *hd* HK file and 'attname' parameter is the attitude file. The output file, (*mkf*), is used to generate GTI based on the parameter values included that are applied at Stage 2 to screen the data. Therefore the filter file has to be generated before starting the Stage 2 for all event modes or the processing of the Imaging mode. NOTE: the filter file in the archive already includes all the specific parameters for the XRT, therefore this step can be ignored and/or used to include new parameters in the file.

4.4 Stage 2: All modes

The second stage involves the screening of the events. The screening criteria can be grouped in three categories:

- those associated with the attitude.
- those associated with the instrument HK parameters

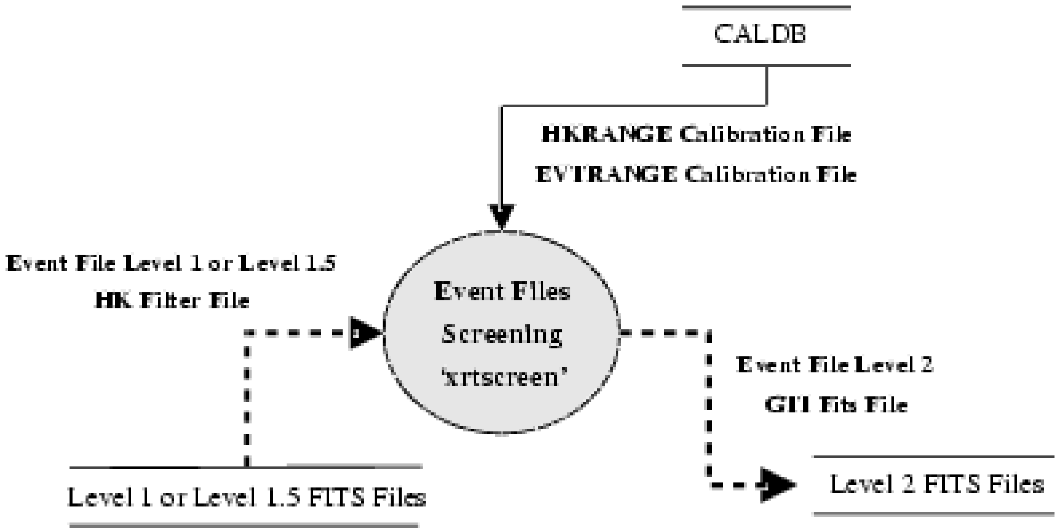

Figure 4.5: The flow diagram for the Stage 2 processing all modes

- those associated with the properties of the events

Chapter 5 gives a description of the different parameters used for the screening. The task `xrtscreen` calculates GTIs based on a boolean expression (which includes the instrument HK and attitude-related parameters) applied on the filter file (*mkf*). Then, it cleans the data using the calculated GTI and a boolean expression that operates on the columns STATUS and GRADE to eliminate all bad events and select on specific grades. The command for the Windowed Timing mode is showed below as example:

```
xrtscreen mkffile=sw00055250019s.mkf.gz createinstrgti=yes
          outfile=sw00055250019xpcw4po_cl.evt createattgti=yes
          infile=sw00055250019xpcw4po_uf.evt gtiscreen=yes outdir=./
          evtscreen=yes gtiexpr=default exprgrade=default expr=default
```

4.5 Stage 1 and 2 : Imaging mode

The processing of the Image mode data involves subtracting the bias, cleaning for bad and saturated pixels, and eliminating the calibration sources. This is done via the `xrtimage` task. To check if an Image exposure is within the nominal setting of instrument HK or attitude parameters, a set of GTI is created using the task `xrtscreen` with the following input parameters:

```
xrtscreen mkffile=sw00113120000s.mkf.gz gtiscreen=no createinstrgti=yes
          createattgti=no outfile=none obsmodescreen=yes outdir=./ evtscreen=no
          expr=none gtiexpr="DEFAULT" gtifile=sw00113120000xim_rwclgti.fits
          infile=sw00113120000xim_rw.img.gz
```

The GTI output can be used in input to `xrtimage` to exclude an exposure that does not satisfy the good HK or attitude setting and while calibrating the image. The basic command to run `xrtimage` uses the following input parameters :

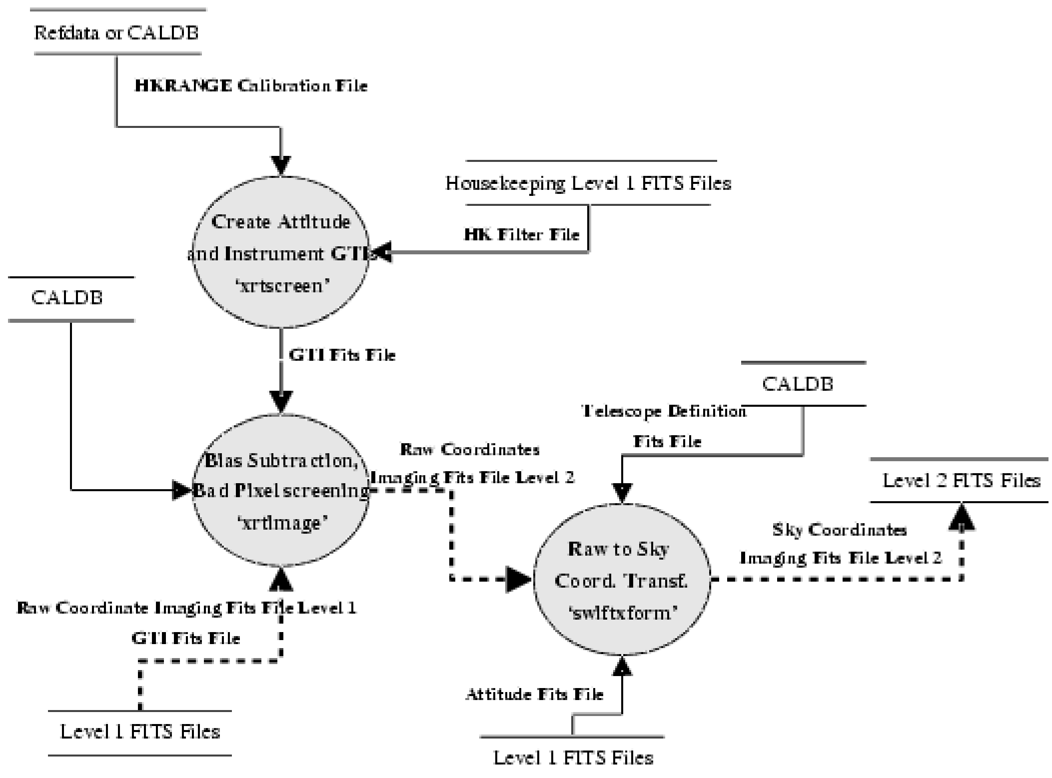

Figure 4.6: The flow diagram for the processing of the Imaging mode

```
xrtimage outfile=sw00113120000xim_rwcl.img infile=sw00113120000xim_rw.img.gz
         gtifile=sw00113120000xim_rwclgti.fits gtiscreen=yes
```

If the parameter gtiscreen is set to 'no' the screening for the GTI is not done and all the Image exposures within a file are processed. If instead is set yes, the task checks that each Image exposure is fully included in the GTI and if not this is not processed and not included in the output file of `xrtimage`

Since only few exposures are taken with the Image mode, if users want to retain all exposures without checking if these fall in a good time interval, `xrtimage` should be run with gtiscreen=no. Next step is to transform the images, from RAW to SKY coordinates. The task `swiftxform` calculates this transformation and the basic command uses the following input parameters :

```
swiftxform infile=sw00070915001xim_rwcl.img to=SKY method=DEFAULT
           outfile=sw00070915001xim_sk.img  ra=247.84 dec=2.15
           attfile=sw00113120000sat.fits.gz teldeffile=swx20010101v004.teldef
```

Note that the pixel values in the images are not counts but instead correspond to the total integrated charge detected in the pixel.

4.6 Calculating the attitude corrected for the TAM

The Telescope Alignment Monitor (TAM) measurements provide a fine correction to the coordinates and improve their accuracy. The TAM correction is applied to the attitude, by using the task 'xrttam'. The basic command uses the following input parameters:

```
xrttam hdfile=sw00113120000xhd.hk.gz  outdir=./ outattfile=DEFAULT
       attfile=sw00113120000sat.fits.gz  outtamfile=DEFAULT
```

The output of *xrttam* is the attitude file corrected by the TAM. To improve the coordinates accuracy for the modes with imaging information, the attitude corrected for the TAM is input to *coordinator* which calculates the sky X and Y coordinates of the event, instead of the normal attitude.

4.7 How to run xrtpipeline

All the above steps have been implemented in the *xrtpipeline* script. The script allows the user to set several parameters, many of which have been already set to a default. *xrtpipeline* assumes that users have downloaded the data from the archive and it searches in the sequence directory for the appropriate input files required to run the individual tasks as listed in the previous sections. The typical directory structure of the data in the archive is the following :

```
/auxil   /xrt  /bat /uvot /log /tdrss
          |
/event /image /hk /products
```

xrtpipeline uses files that are located in the `auxil` and `xrt` directories. Specifically, from the `auxil` directory it uses the attitude file, the two line element file and/or the make filter file. From the `xrt` directory, it uses the Level 1 events and image files in the `/event` and `image` directories respectively, and the header file located in the `/hk` directory. The outputs of the *xrtpipeline* consists of Level 1 files re-calibrated according to the CALDB files in use, and Level 2 files, e.g. screened events, for each of the read out modes and each of the observing modes (slew, pointing or settling) included in an observation. It also outputs standard lightcurve, spectra and arfs for PC, WT, and PD data for the targets (in FITS and GIF) and sky images of the full field of view for Image, Windowed Timing and Photon Counting mode (in FITS and GIF). By default, *xrtpipeline* also generates the make filter file used for screening.
Note: the spectra and lightcurves within *xrtpipeline* are derived using *xselect* on a default region for the Photon Counting and Windowed Timing data. All products for all modes are not background subtracted.
The input parameters of *xrtpipeline* include all of those necessary to run the tasks listed in this chapter as well as parameters needed to run subsets of the data processing stages. The important parameters needed to start the data reduction at different stages are the 'entrystage' and 'exitstage'. The values for these parameters range from 1-3 and they reproduce the stages shown in figure 4.1. The value for 'entrystage' should always be higher than the value of 'exitstage'. By setting these parameters, it is possible for example to apply different screening criteria to the calibrated event files without re-running the entire processing. Another possible use of these parameters is the generation of the Level 2 files which can than be used by xselect to extract source spectra, lightcurves with user-defined settings. Listed below are six examples of how to run *xrtpipeline*.

1. The basic *xrtpipeline* command uses the following required input parameters:

   ```
   xrtpipeline indir=./00035020003 outdir=./output steminputs=sw00035020003
               srcra=OBJECT srcdec=OBJECT
   ```

where 'indir' is the directory name where the data files are located, 'outdir' is the output directory (this must exist on disk), 'steminputs' is the common root for the filenames and 'srcra' and 'srcdec' are the source RA and Dec. By setting the values of srcra and srcdec to OBJECT the code looks at the keywords RA_OBJ and DEC_OBJ for the source coordinates and RA_PNT and DEC_PNT for the pointing position present in the event files.

2. The following command shows how to set the pointing position to a non default value by setting the parameters 'pntra', 'pntdec' and 'roll'. This is useful when for example the pointing position is calculated for a time interval different than the standard one listed in the RA_PNT and DEC_PNT keywords.

```
xrtpipeline indir=./00035020003
            steminputs=sw00035020003 outdir=./output srcra=OBJECT
            srcdec=OBJECT pntra=217.14 pntdec=42.68 pntroll=0.0
```

3. The next command shows how to specify source coordinates different than the default, obtained for example by different measurements or from a catalog, by setting the parameters 'srcra' and 'srcdec'. In addition, since the parameter 'createmkffile' is set to 'no', *xrtpipeline* does not derive the make filter file but uses the one already available in the `/auxil` directory. Finally, the images extracted during the processing are plotted on the screen since the parameter 'display' is set to 'yes' (default is 'no').

```
xrtpipeline indir=./00035020003
            steminputs=sw00035020003 outdir=./output srcra=217.135978237
            srcdec=42.6717438456 display=yes createmkffile=no
```

4. In this example *xrtpipeline* run a different grade selection on the Level 1 files created in a previous run of *xrtpipeline* and generates new Level 2 and 3 files. This is achieved by setting the parameters 'entrystage' and 'exitstage' to 2 and 3 respectively. The new screening criteria, the grades in this example, are set in the parameters 'exprpcgrade' 'exprwtgrade' 'exprpdgrade' and they are applied to the Level 1 files found in the directory 'output' since 'evtfilesfromarchive' set to 'no'.

```
xrtpipeline indir=./00035020003 steminputs=sw00035020003 outdir=./output
            srcra=217.135978237 srcdec=42.6717438456 entrystage=2 exitstage=3
            evtfilesfromarchive=no exprpcgrade="0" exprwtgrade="0-2"
            exprpdgrade="0-2" clobber=yes
```

5. In this example, only the Low-rate Photodiode mode data obtained during pointing mode for time intervals when the CCD temperature was between -102 and -47 are processed. The mode is selected via the parameter 'datamode', where the possible value are: lr, Low-rate, pu, Piled-up, pc, Photon Counting, wt, Windowed timing, and im, Image. The expression on the temperature is specified in the parameter 'gtiexpr' and is applied to the makefilter file that generates the good time intervals. The parameter 'obsmode' allows the user to select data taken during 'POINTING' (as in the example), 'SETTLING' (within 10 arcmin of the source) or the 'SLEW'.

```
xrtpipeline indir=./00035020003 steminputs=sw00035020003 outdir=./output
            srcra=OBJECT srcdec=OBJECT obsmode=POINTING  datamode=lr
            gtiexpr="CCDTemp>=-102&&CCDTemp<=-47"
```

6. This example shows how to use 'cleancols' and another screening parameter,'obsmodescreen'. The first if set to 'no' does not copy the optional columns from Level 1 to the Level 2 files. The second acts on the make filter file that defines whether the spacecraft is in slew, within 10 arcmin of the source or in pointing position.

   ```
   xrtpipeline indir=./00035020003 steminputs=sw00035020003 outdir=OUTDIR
               srcra=OBJECT srcdec=OBJECT obsmodescreen=no cleancols=no
   ```

   When 'obsmodescreen' is set to 'yes', the *xrtpipeline* adds to the expression used to generate GTIs, the screening related to the observation mode. These are :

   ```
   "SETTLED==0&&TEN_ARCMIN==0"
   ```

   to select data taken during slew;

   ```
   "SETTLED==0&&TEN_ARCMIN==1"
   ```

   to select data taken in 10 arcmin from the source, and `"SETTLED==1"` to select data in pointing)

Chapter 5

SCREENING CRITERIA

5.1 Introduction

The *Swift* XRT data as collected on-board include 'bad' events which have to be rejected in the course of data analysis, because they are for a number of reasons inappropriate for scientific analysis. This chapter lists the criteria and the methods that can be used to filter out these data. Standard data screening is carried out during processing at SDC, and the cleaned events lists (Level 2) are placed in the archive. The standard screening criteria have been defined by the XRT instrument team to provide a good balance between rejecting bad events without compromising signal-to-noise, and the convenience of providing users with a manageable set of criteria.
Although normally these standard cleaned event lists are used to extract images, light-curves and spectra, users may want to apply their own data screening criteria, which may be looser or tighter than the standard ones. In this case, users needs to understand the kind of data screening that should normally be applied to the data before extracting the high-level scientific products.
The screening criteria are set by selecting on parameters that are either present in the make filter (`mkf`) file or in columns in the event file. To apply different screening criteria compared to the standard ones, users have to start from the Level 1 data for Photon Counting Mode and the Level 1a data for Photodiode and Windowed Timing mode using the FITS files that can be taken from the archive.
Alternatively, to tighten the range of values for the parameters used in the standard screening criteria users can start from the Level 2 files. The new screening criteria are applied to the data using `xrtscreen`, `xrtpipeline` or `XSELECT` software tasks and may be specified using a Fortran-like style (e.g., ELV.gt.5) or a C-like style (ELV > 5).

5.2 Screening Criteria associate with the ACS

One set of screening criteria are obtained by considering parameters related to the satellite position. These parameters are included in the `mkf` and are used to generate GTIs where their values are within specified ranges. During the PV phase the following set of parameters were found to be effective when screening the data on several observations. These parameters are part of the standard screening in `xrtpipeline`, but they can be overwritten by setting a new experession in the `xrtpipeline` parameter 'gtiexpr' (see examples).

- The Elevation angle, **ELV** and Bright Earth **BR_EARTH**, the first is the angle between the Earth's limb and the pointing direction, the second is an angle between the pointing direction of the satellite and the day-night terminator. Data are excluded when the **ELV** is lower than 45 deg and the **BR_EARTH** is lower than 120.

- The **ANG_DIST** is the angular distance in deg between the nominal pointing position and the pointing vector. Data should be included using an angular distance of 0.08 deg.
- The Sun Angle, **SUN_ANGLE** is the angle in deg between the sun center and the pointing direction. Its value should be always bigger than 45 deg.
- The Moon Angle, **MOON_ANGLE** is the angle in deg between the moon center and the pointing direction. Its value should be always bigger than 30 deg.

The names in **bold** are the column names in the `mkf` file. During the mission, the ranges for these parameters will be optimized and the standard values will be documented in this chapter. The South Atlantic Anomaly (SAA) has not been included since on-board the instrument do not collect data when enter the SAA if is in Auto state. Data can be collected during the SAA only if the instrument is in Manual State typically used during calibration, or if the spacecraft slews on a GRB during an SAA. In the latter case data are collected in Image mode first and after of few ms of Pileup Photodiode. In this case user do not want screen by the SAA but the data need to be carefully evaluated.

5.3 Screening Criteria Specific to the XRT

There are two groups of screening parameters which are specific to the XRT. The first is associated with the instrumental parameters (HK) and the second with characteristics of the events themselves.

5.3.1 Instrument Parameters

Screening on specific instrument parameters is done to ensure that data are always included within certain specific boundaries which are considered the best for the instrument's performance. The standard expressions used are listed below for each of the HK parameters considered:

- CCDTemp >= -102 && CCDTemp <= -47 ; where CCDTemp is the temperature of the CCD
- Vod1 >= 29.82 && Vod1 <= 30.25 ; where Vod1 is the Output Drain Voltage for Amp 1 (left Amp)
- Vod2 >= 29.30 && Vod2 <= 29.80; where Vod2 is the Output Drain Voltage for Amp 2 (right Amp)
- Vrd1 >= 16.40 && Vrd1 <= 16.80; where Vrd1 is the Reference Voltage for Amp 1
- Vrd2 >= 16.45 && Vrd2 <= 16.90; where Vrd2 is the Reference Voltage for Amp 2
- Vsub >= -0.1 && Vsub <= 0.1; where Vsub is the Substrate Bias Voltage
- Vbackjun >= -0.1 && Vbackjun <= 0.1; where Vbackjun is the Back Junction Bias Voltage
- BaseLin1 >= 0.1 && BaseLin1 <= 0.4; where Baseline1 is the Baseline Voltage for Signal Chan A

- BaseLin2 >= -0.1 && BaseLin2 <= 0.1; where Baseline2 is the Baseline Voltage for Signal Chan B

These parameters are listed in the mkf file. The screening criteria on these HK parameters together with those associated with the ACS are used to generate GTIs for when the events are considered 'good'. NOTE: The above parameters are not expected to change but for the temeprature. It is in fact recommended to always plot the temperature values versus time to access the overall temperature behaviour during on observation. This can be achived by using *fplot* or *fv* either with the **mkf** file or with the XRT **xhd.hk** and plot the column CCDTemp versus TIME.

5.3.2 Event characteristics

The current XRT specific screening criteria are :

- Removal of calibration sources
- Removal of bad pixels (dead, hot and flickering) and pixels below the event threshold for earth limb screening
- Removal of saturated pixels
- Grade selection
- Removal of partially exposed pixels (photodiode mode)
- Removal of telemetry saturated frames

Calibration Sources, Bad Pixels and Events Above Threshold

When the XRT is operated in Photon Counting mode, the data telemetered down can include the four corner calibration sources if the XRT was running in full window configuration. The pixels associated with the calibration sources are flagged with special values in the STATUS column.
When the XRT is operated in Photon Counting or Windowed Timing modes, bad pixels or bad columns can be identified and flagged with special values in the STATUS column.
While for all modes except for the Pileup Photodiode only data within threshold boundaries are telemetered down, on the ground when the events are reconstructed, it is possible to "create" an event above threshold or a saturated one for which the PHA value is above 4095. These events are flagged and a special value set in the STATUS column.
For Photon Counting it is possible to increase on ground the event threshold, to further exclude effect due to the bright earth, by setting within xrtflagpix the threshold for the central pixel higher than the lower level discriminator, set on-board at 80. The expression used to remove calibration sources, bad pixels, central events below the threshold and saturated pixels is

STATUS == b0

Grade Selection

The grade is a description of the shape of the charge spread created by an event in a CCD. Certain grades have better spectral resolution than others (grade 0 has the best resolution) or are more likely to be X-ray events. There are 32 grades defined for the XRT (as described in chapter 3) for the Photon Counting mode and 15 for the Photodiode and Windowed Timing modes. The standard grade selection for each of the modes is :

- Photon Counting
  GRADE >= 0 && GRADE <= 12
- Photodiode
  GRADE >= 0 && GRADE <= 5
- Windowed Timing
  GRADE >= 0 && GRADE <= 2

Partially Exposed Pixels, Telemetry-Saturated Frames or Piled-up Frames

These are special categories of screening that are applied when generating the Level 1a from the Level 1 in Photodiode. They are listed here for completeness and it is foreseen that users would not have to screen for these criteria.

In the Photodiode mode, at the start of the first frame there are events that are not fully exposed. These events must be eliminated before the event reconstruction takes place, otherwise there will be an incorrect assignment of the grade. This is achieved by calculating appropriate event GTIs with the 'xrttimetag' task.
The same task also calculates the GTI to exclude telemetry-saturated frames that should not be used to obtain flux estimates. The GTIs, excluding both partially exposed pixels and telemetry-saturated frames, are used to filter out the events when creating the Level 1a data files.

5.4 Summary

The following is a summary of the typical screening criteria that should be applied to XRT data:

1. Removal of bad pixels (dead, hot and flickering) and pixels below the event threshold for earth limb screening
2. Remove calibration sources
3. Select grades in the range of
   - 0-12 for the Photon Counting
   - 0-2 for Windowed Timing
   - 0-5 for Photodiode
4. Select events based on a minimum elevation angle (ELV) and bright earth (BR_EARTH)
5. Select events based on angular distance (ANG_DIST)
6. Select events based on minimun sun and moon angle (SUN_ANGLE and MOON_ANGLE)
7. Select events corresponding to housekeeping parameter values included in the fixed ranges
8. Remove partially exposed events (Photodiode and Window Timing) and saturated frames (Photodiode).

5.5 How to Screen the Data

There are three methods that allow the user to perform customized screening. These are implemented with the following programs:

- **xrtscreen**
  This is a perl script that runs all the XRT tasks and allows the user to specify screening criteria via the following input parameters: 'gtiexpr', 'exprgrade', and 'expr'. The script uses *XSELECT* and/or fselect to screen the event list. If all the default criteria are selected in `xrtscreen`, the result will be a set of cleaned events lists identical to those produced by the standard SDC processing.

- **xrtpipeline**
  This is a perl script that runs all the XRT tasks, including `xrtscreen`, and, starting from the Level 1 found in the archive recreates all the other data files. If an observation contains data in different modes or mode settings, they will all are processed. Within `xrtpipeline`, the screening criteria are specified via the following input parameters: 'gtiexpr', 'exprpcgrade', 'exprpdgrade', 'exprwtgrade', 'exprpc', 'exprpd' and 'exprwt'.

- **XSELECT**
  The expressions specified in the `xrtscreen` or `xrtpipleine` screening criteria parameters can be replicated and applied by running `XSELECT` in stand-alone mode (which is in fact the main workhorse used by the above scripts).

5.5.1 Example of How to Use xrtscreen

Let us assume that, after examining the Level 2 data taken with the Photon counting mode already cleaned by the standard criteria, the user decides to select only event with grades in the range of 0-2 and a different cut-off rigidity which is tighter that the standard one. These new criteria can be specified in `xrtscreen` by:

```
xrtscreen mkffile=sw00055250019s.mkf createinstrgti=yes outdir=./
          outfile=sw00055250019xpcw4po_cl_new.evt evtscreen=yes
          createattgti=yes infile=sw00055250019xpcw4po_cl.evt gtiscreen=yes
          gtiexpr="ANG_DIST<0.08" exprgrade=0-2 expr=DEFAULT
```

where the criteria on the cut-off rigidity are translated into GTIs by looking at the COR_SAX column listed in the `mkf` file. The GTI are applied by setting the 'gtiscreen' parameter to yes. The grade selection in contrast is done by selecting on the column GRADE in the event file and is applied by setting the parameter 'evtscreen' to yes.

5.5.2 Example of How to Use xrtpipeline

When using *xrtpipeline*, users have to start from the Level 1 file. Following the example above the command will be

```
xrtpipeline srcra=171.1014 srcdec=14.2305 indir=../00055250019 outdir=./
            steminputs=sw00055250019 gtiexpr="ANG_DIST<0.08" exprpcgrade=0-2
            expprpc=DEFAULT
```

Note that in this case the `mkf` file is generated within the script and the Level 2 file of the Photon Counting mode will be screened only for these two criteria and by the default setting for the parameter 'expprpc' which applies the screening to the STATUS column.

5.5.3 Example of how to use XSELECT

Using `XSELECT`, it is possible to screen the data interactively and also to extract the high-level products, as shown in the next chapter.

Within `XSELECT`, it is possible to reproduce the screening expression specified in `xrtscreen` via the parameters 'gtiexpr','exprgrade', and 'expr' by using the `XSELECT` commands '`select mkf`', '`select event`' and '`filter grade`'. The first command allows the user to specify an appropriate boolean expression and the desired ranges of values for generating a GTI from the parameters listed in the mkf file. For example:

```
xsel:Swift-XRT > select mkf "ANG_DIST<0.08&&CCDTemp>-102&&CCDTemp<-50"
> Enter the filter file directory >[.]
```

The second command allows the user to select on the event STATUS column for the expression used in the `xrtscreen` parameters 'expr'. For example :

```
xsel:Swift-XRT > select event "status==b0'
```

The last command, 'filter grade', allows the user to select the event file for a specific range of grades (e.g. 'filter grade 0-2, 4' select grade 0, 1,2 and 4).

The above example can be reproduced within `XSELECT` using the following commands :

```
==========================================================================================
                         **  XSELECT V2.2a  **

> Enter session name >[xsel]
xsel:ASCA > read events sw00055250019xlrb1po_cl.evt
> Enter the Event file dir >[.//]
Got new mission: SWIFT
> Reset the mission ? >[yes]

 Notes: XSELECT set up for      SWIFT
 Time keyword is TIME       in units of s
 Default timing binsize =   1.0000

Setting...
 Image  keywords   = DETX       DETY        with binning =    1
 WMAP   keywords   = DETX       DETY        with binning =    1
 Energy keyword   = PI                     with binning =    1

Getting Min and Max for Energy Column...
Got min and max for PI:     0   1023

Got the minimum time resolution of the read data:  0.14000E-03
```

```
MJDREF =  5.1910000742870E+04 with TIMESYS = TT
 Number of files read in:  1

******************* Observation Catalogue *******************

Data Directory is: /local/swift/00055250019_out/
HK Directory is: /local/swift/00055250019_out/


        OBJECT      OBS_ID      DATE-OBS    DATAMODE
      1 PG1121+145  00055250019 2005-04-05T LOWRATE

xsel:SWIFT-XRT-LOWRATE > select mkf "ANG_DIST<0.08" mkf_name=sw00055250019s.mkf mkf_dir=./
xsel:SWIFT-XRT-LOWRATE > filter grade 0-2
xsel:SWIFT-XRT-LOWRATE > extract events
=================================================================================================
```

where the 'select' and 'filter' commands set the screening criteria and 'extract' applies these screening criteria to the event file.

Chapter 6

Extraction of Products

6.1 Introduction

This chapter describes how to extract the high-level products from the Level 2 screened event files using XSELECT. The following products can be extracted for the different XRT modes:

- Photon Counting : 2D Image, Lightcurve, Spectrum
- Windowed Timing : 1D Image, Lightcurve, Spectrum
- Photodiode : Lightcurve and Spectrum

The following sections show how to use XSELECT, set filters, and give examples for how to extract products and prepare the data for further analysis.

6.2 Using XSELECT

For each XRT mode, XSELECT has a number of default settings in place that users may change via the XSELECT command 'set'. The default settings include the names of the columns to extract the spectrum, the weighed map (WMAP) placed in the primary header of the spectrum, the image and a default binning for the lightcurve which can be different from time resolution of the data. For example, when reading the Photon Counting mode data, the following defaults are set and shown on the screen :

```
> xselect

                         **  XSELECT V2.2a  **

> Enter session name >[xsel]
xsel:ASCA > read events sw00050300005xpcw4po_cl.evt
> Enter the Event file dir >[.//]
Got new mission: SWIFT
> Reset the mission ? >[yes]

 Notes: XSELECT set up for      SWIFT
 Time keyword is TIME       in units of s
```

```
 Default timing binsize =   5.0000

Setting...
 Image  keywords   = X          Y           with binning =    1
 WMAP   keywords   = X          Y           with binning =    1
 Energy keyword   = PI                     with binning =    1

Getting Min and Max for Energy Column...
Got min and max for PI:     0   1023

Got the minimum time resolution of the read data:   2.5073
MJDREF =  5.1910000742870E+04 with TIMESYS = TT
 Number of files read in:  1

******************** Observation Catalogue ********************

Data Directory is: /local/swift/
HK Directory is: /local/swift/


        OBJECT      OBS_ID      DATE-OBS    DATAMODE
      1 Mrk876      00050300005 2005-01-26T PHOTON

xsel:SWIFT-XRT-PHOTON >
```

Thus as the default, the image is created using the columns X and Y of the event file; the spectrum is obtained from the PI column, the WMAP stored in the primary of the spectrum is from the X and Y columns and the default lightcurve binning is set to 5 seconds.
The default settings for the "Timing Modes" differ in the standard lightcurve binning set to 1 second. The default image and WMAP coordinates for the Photodiode Mode are both set to DETX and DETY, but this mode does not have image capability and the coordinates correspond to the source position on the detector.

To extract products, the `XSELECT` command '`extract`' is used, followed by the product type (e.g. '`extract image`', '`extract curve`', '`extract spectrum`'), or '`extract all`' creates an image, a spectrum and a lightcurve. The `XSELECT` command '`save`' followed by the product type and a filename writes the selected products to files, or '`save all`' writes the latest products extracted an image, a spectrum and a lightcurve simultaneously to separate files sharing a root filename. Using the example above :

```
xsel:SWIFT-XRT-PHOTON > extract all
extractor v4.47      1 Dec 2004
 Getting FITS WCS Keywords
 Doing file: /local/swift/sw00050300005xpcw4po_cl.evt
100% completed
          Total      Good    Bad: Region      Time     Phase     Grade     Cut
           4772      4772              0         0         0         0         0
===============================================================================
   Grand Total      Good   Bad:  Region      Time     Phase     Grade    Cut
          4772      4772              0         0         0         0         0
   in  12040.     seconds
```

```
 Fits light curve has     4772 counts for 0.3963     counts/sec
 Thresholding removed more than half the bins
 Try exposure=0.0 on the extract command in xselect
 or lcthresh=0.0 if running extractor stand-alone
 Spectrum         has     4772 counts for 0.3963     counts/sec
 ... written the PHA data Extension
 Image            has     4772 counts for 0.3963     counts/sec
xsel:SWIFT-XRT-PHOTON > save all source_pc

Saving the Spectrum:
Wrote spectrum to source_pc.pha

Saving the Image:
Wrote image to file source_pc.img

Saving the Light Curve:
Wrote FITS light curve to file source_pc.lc
xsel:SWIFT-XRT-PHOTON >
```

Any products extracted as in the above example do not have any data filtering applied. Within `XSELECT` it is possible to set and apply the following types of filters: region (positional), grade, time, energy, intensity and phase. For the XRT modes with imaging capability (Photon Counting and Windowed Timing), it is recommended to filter the data for the region containing the source. The other types of filters are optional and depend on the desired type of analysis. The following section shows how to apply these various filters.
To extract background information, the procedure is different for each of the modes. In Photon Counting mode, the background should be selected from a source free region seen in the image. For the Windowed timing mode, the source should appear in the middle of the 1-D image, therefore the background can be selected using in regions on either site of the source. In Photodiode mode the background should instead be derived from the data acquired during the slew.

6.3 Setting Filters

The `XSELECT` command '`filter`' allows the user set the different type of filters, and these are then applied to the data using the '`extract`' command.
It is possible to enter multiple filters, for example a region, time and intensity filter, and all three filters will be applied the next time that the '`extract`' command is invoked. It is also possible to remove a filter with the command '`clear`' followed by the filter type. For example '`clear region source.reg`' removes the region filter named '`source.reg`'.
The command '`show filters`' allows the user to check what filters are in place, and it is recommended to do this before using the '`extract`' command.

6.3.1 Grade Filtering

The XRT Level 2 event files are screened for a set of recommended grades. However, users may want to further restrict this selection to specific grade settings. The selection is applied by testing on the GRADE column in the event file. The `XSELECT` command to enter a selection on grade is '`filter grade`' followed by a single grade number (e.g. 0), a range (e.g. 0-2), an upper limit (e.g. < 4), or a lower limit (e.g. > 3). Example:

```
filter grade 0

filter grade 0-2
```

where the first command selects only event with grade 0, and the second command select events with grades in the 0-2 range. Users should be awarded that the XRT response matrices are made for specific grade selection, therefore this selection cannot be arbitrary. By default, the XRT Level 2 file contain events with the following grade selection: 0-12 for the Photon Counting mode, 0-5 for the Photodiode mode and 0-2 for the Windowed Timing mode. Response matrices are also available for the following grade selection: 0 for all three modes, 0-2 for the Photodiode mode and 0-4 for the Photon Counting mode.

6.3.2 Region Filtering

To create a region filter, users have first to examine the image and then select the region so as to include the source of interest. This type of filter applies only to the Photon Counting and Windowed Timing XRT modes, since only these modes have position information. For the Photodiode, mode no region selection is necessary. The region files can be created within `XSELECT` with the following steps:

1. Users should decide whether they want to work in SKY coordinates (default) or DETECTOR coordinates and change the coordinates setting using the command '`set image sky`' or '`set image detector`' for SKY and DETECTOR, respectively.

2. Extract an image from the event file read in `XSELECT` using the '`extract image`' command.

3. Plot the image with the '`plot image`' command which invokes `ds9`.

4. Create the region file within '`ds9`' in the usual way, selecting on a region menu the shape of the region, the file format and the file coordinates system. By default, the region file format is set to `ds9` and the coordinates system to 'physical', and this creates a region in pixel space using the original pixel numbering associated with the columns chosen to create the image. Alternatively, by setting the coordinates system to 'WCS' the region files are written using sky coordinates. For a circular shape, examples of the content of the region file for different coordinates system and file format are :

   ```
   fk5;circle(359.99051,0.00098221334,20.849088")   WCS, format DS9
   circle(515,502,8.8444329)                        Physical, format Ciao SAOtng
   ```

   Notes: (1) Region files can also be created with `fv` and `XIMAGE` and imported into `XSELECT`; (2) The procedure to define regions for the background is the same as for the source.

To set the region filter in `XSELECT`, the command is '`filter region region.reg`' where the region.reg is the region file created with `ds9` or other external software. For the Photon Counting mode with 2D positional information, the shape of the region should be a circle for a point-like source. The 90% of the PSF at 1.5 keV is enclosed by a 20 pixels radius circle (corresponding to $\sim$ 47 arcsec) and is weakly dependent on the off-axis angle. An example of the circle region in pixels is therefore :

```
CIRCLE(455.7, 501.2, 20.0)
```

where (455.7, 501.2) are the coordinates of the center of the circle and 20 is the radius of the circle in pixels.

For the Windowed Timing mode, the region shape is instead a box, since only 1D positional information is available along the DETX coordinate. A box region is specified by five parameters: the X and Y center, the width, the height and an angle. The recommended width is 40 pixels, with an height large enough to include all the photons in the Y dimension. The BOX rotation angle should be in degrees and is set to zero when extracting products in detector coordinate and to roll-90 if the products are extracted in sky coordinates (where 'roll' is the spacecraft roll angle expressed in degrees). An example of a box region in pixel using the same center as above and a rotation angle of 50 degrees and a Y-dimension of 30 pixels.

```
BOX(455.7, 501.2, 40.0, 30.0, 50.0)
```

6.3.3 Time Filtering

There are three different methods to select data based on time. These are :

1. Specifying parts of the light curve with the mouse cursor. The selected times are captured in a FITS GTI file. The command is '`filter time cursor`';
2. Entering the start and stop times at the keyboard. The selected times are captured in a FITS GTI file. The command is '`filter time UT`' or '`filter time SCC`' or '`filter time MJD`';
3. Entering a file containing the desired start and stop times. The command is '`filter time file filename`'

The intervals within each file are subjected to a logical OR operation, i.e., an event has to fall within one of the GTIs listed in a file to be accepted as a good event. It is possible to enter time selection using more than one method. In this case the filters are combined using a logical AND operation. An event has to fall within *every* GTI in each time filter file to be accepted as a good event. The time filters can be applied to all XRT modes.

Cursor Time Filtering

To use the cursor method to select time, users first have to extract a lightcurve and then use the command '`filter time cursor`'. The lightcurve is plotted together with a set of instruction to use the cursor to select part of the lightcurve. Leaving the cursor selection, a GTI temporary file is created containing the selected times, which will be used when the next '`extract`' command is issued. The effect of this filter can be removed with the '`clear time cursor`' command. The temporary file can be saved using the command '`save time cursor filename`', otherwise it is deleted upon exiting `XSELECT`.

Keyboard Time Filtering

The keyboard method allows users to enter the start and stop time from the keyboard. The times can be specified with different formats :

1. **Universal Time** (UT), in which case the command is '`filter time ut`' and the format for the time is, e.g., 2003-12-18T17:18:05

2. **SpaceCraft Clock Time** (SCC), in which case the command is '`filter time scc`' and the format for the time is, e.g., 9.372975399487495E+07 or 93729753.99487495.

3. **Modified Julian Day** (MJD), in which case the command is '`filter time mjd`' and the format for the time is, e.g., 52991.66012956046

The selected times are written to a GTI temporary file and will be used next time the '`extract`' command is issued. The effect of this filter can be removed with the '`clear time cursor`' command. The temporary file can be saved using the command '`save time keyboard filename`', otherwise it is deleted when exiting `XSELECT`. If a lightcurve was previously extracted, entering the command, '`filter time scc`' will cause the light curve to be plotted and the selection is via the cursor (as for the cursor time filtering).

Time filtering from files

This method allows users to enter a filename (either ASCII or FITS), containing the start and stop time necessary to filter the data, with the command '`filter time file filename`' where filename is the name of the file containing the start and stop times. The ASCII file contain pairs, one per line, of the start and stop times, specified as Space Craft clock time (units are seconds) , as for example :

```
9.372975399487495E+07   9.374917324292182E+07
9.376167806128120E+07   9.378212711987495E+07
```

The FITS file has the GTI format. To remove the effect of this time filter, the command '`clear time ASCII`' removes any ASCII GTI file and '`clear time fits`' removes any FITS GTI file.

6.3.4 Energy Filtering

Energy filters are useful ways to extract lightcurves or images for specific energy bands. The XRT event files do not have an energy column but to each photon a PHA and PI channel is assigned, and the energy selection is via these columns. To set an energy filter, users have to first define the energy column, and then input a lower and upper channel range. This is achieved by the following commands:

- `set phaname PI`, which sets the PI column as the energy column.
- `filter pha_cutoff 100-700`, which filters all data within the channel range 100-700 of the PI column.

Note: for the XRT the spectral information is by default extracted from the PI column, since its values are corrected for temporal and temperature changes in the gain and for positional gain variation (Charge Transfer Inefficiency, CTI). In PI space the energy-to-channel conversion is approximately linear and is about PI = 100*E, where the E is the the energy in keV. The energy filters can be applied to all XRT modes.

6.3.5 Intensity Filtering

To define an intensity filter, users have to first extract and plot a lightcurve and then use the command '`filter intensity min-max`' where min and max is the range of intensity selected.

This filter generates a temporary GTI file where each interval corresponds to times where the intensity is included in the specified range. The filter is used next time the '`extract`' command is issued. The temporary file can be saved using the command '`save time keyboard filename`', otherwise it is deleted upon exiting from `XSELECT`. The effect of this filter can be removed with the '`clear intensity`' command. Please note that any energy filters used to create the lightcurve will be applied in conjunction with the intensity filter the next time the '`extract `' command is issued, unless they are cleared before extracting the product. If, for example, the intensity filter is created using selections made based on a light curve in the soft energy band, the energy filter should be cleared before extracting the spectrum or else all events in the soft energy band will be filtered out.

6.3.6 Phase Filtering

To set a phase filter, users have to specify an epoch, a period and a phase range using the command '`filter phase epoch period ph1-ph2`'. For example :

```
filter phase 52991.66012956046 0.00782407 0.0-0.2
```

will cause events to be extracted between the phases of 0 and 0.2, where the full range in phase is 0-1. The first argument is the epoch of phase zero in MJD, while the second argument is the period in days (676 seconds, in this case). The command `clear phase` removes a phase filter.

6.4 Examples

6.4.1 Extract spectra for Photon Counting mode data

The example below is an `XSELECT` log of the commands used to generate a spectrum for the source and background in the Photon Counting mode. The image generated in sky coordinates (the default setting is the X and Y columns) is also saved.

- Start-up `XSELECT` and read a Photon Counting mode data file
- Extract an image and save to a file
- Display the image
- Select the regions for the source and the background in ds9 (the commands for the selection in ds9 are not shown, but the results are in Fig 2.1)

```
===============================================================================
> xselect

                         **  XSELECT V2.2a  **

> Enter session name >[xsel]
xsel:ASCA > read events sw00050300005xpcw4po_cl.evt
> Enter the Event file dir >[.//]
Got new mission: SWIFT
> Reset the mission ? >[yes]
```

```
 Notes: XSELECT set up for      SWIFT
 Time keyword is TIME       in units of s
 Default timing binsize =   5.0000

Setting...
 Image  keywords   = X          Y           with binning =    1
 WMAP   keywords   = X          Y           with binning =    1
 Energy keyword   = PI                     with binning =    1

Getting Min and Max for Energy Column...
Got min and max for PI:     0   1023

Got the minimum time resolution of the read data:   2.5073
MJDREF =  5.1910000742870E+04 with TIMESYS = TT
 Number of files read in:  1

******************** Observation Catalogue ********************

Data Directory is: /local/swift/
HK Directory is: /local/swift/


        OBJECT      OBS_ID      DATE-OBS    DATAMODE
      1 Mrk876      00050300005 2005-01-26T PHOTON

xsel:SWIFT-XRT-PHOTON > extract image
extractor v4.47      1 Dec 2004
 Getting FITS WCS Keywords
 Doing file: /local/swift/sw00050300005xpcw4po_cl.evt
100% completed
          Total      Good    Bad: Region      Time     Phase     Grade     Cut
           4772      4772              0         0         0         0         0
===============================================================================
   Grand Total      Good   Bad:  Region      Time     Phase     Grade    Cut
          4772      4772              0         0         0         0         0
   in  12040.     seconds
 Image            has     4772 counts for 0.3963     counts/sec
xsel:SWIFT-XRT-PHOTON > save image image_pc.img
Wrote image to file image_pc.img
xsel:SWIFT-XRT-PHOTON > plot image
===============================================================================
```

After the region has been selected :

- Set the source region filter
- Extract source spectrum
- Save the source spectrum to a file

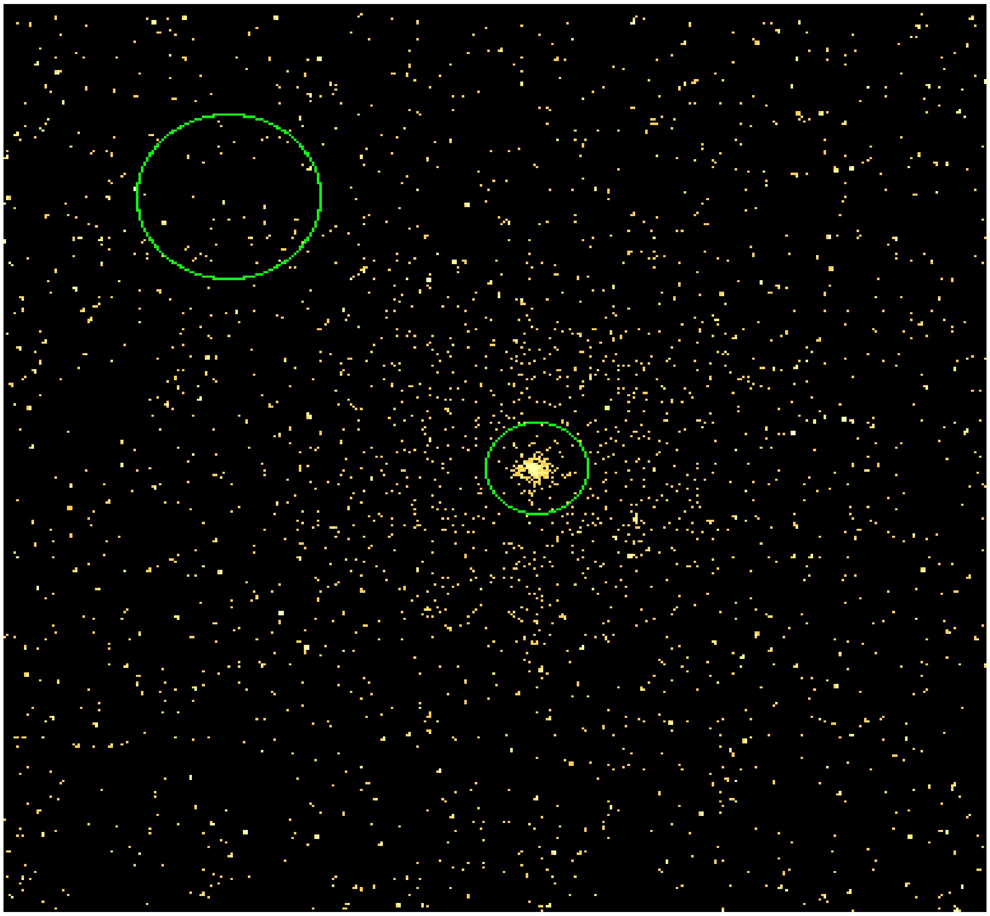

Figure 6.1: Image Plot with the source and background region selected

- Remove filter source region file
- Set the background region filter
- Extract background spectrum
- Save a background spectrum to a file
- Remove filter background region file

```
===============================================================================================
xsel:SWIFT-XRT-PHOTON > filter region source.reg
xsel:SWIFT-XRT-PHOTON > extract spectrum
extractor v4.47      1 Dec 2004
 Getting FITS WCS Keywords
 Doing file: /local/swift/sw00050300005xpcw4po_cl.evt
100% completed
          Total      Good    Bad: Region      Time      Phase      Grade      Cut
           4772      2331           2441         0          0          0          0
===============================================================================
   Grand Total      Good   Bad:  Region      Time      Phase      Grade     Cut
          4772      2331           2441         0          0          0          0
   in  12040.     seconds
```

```
 Spectrum         has     2331 counts for 0.1936     counts/sec
 ... written the PHA data Extension
xsel:SWIFT-XRT-PHOTON > save spectrum src_pc.pha
Wrote spectrum to src_pc.pha
xsel:SWIFT-XRT-PHOTON > clear region
xsel:SWIFT-XRT-PHOTON > show status

              *** Status of XSELECT ***

Plot device is /XS

 *** MISSION ***

SWIFT NONE XRT PHOTON
Time keyword is TIME       in units of s
Default timing binsize =   5.0000
Minimum timing resolution:   2.5073
Energy keyword is PI         with binning     1
Max and min for PI        :     0    1023
 Image keywords   = X          Y           with binning =    1
 WMAP  keywords   = X          Y           with binning =    1

 *** DATA ***

The data directory is /local/swift/

The obscat xsel_read_cat.xsl has been made.

One data file has been read in.
GTI files are in use.

 *** PRODUCTS ***

 A spectrum has been accumulated.
 An image has been accumulated.

 *** SELECTIONS ***

     NONE

 *** FILTERS ***

     NONE

xsel:SWIFT-XRT-PHOTON > filter region back.reg
xsel:SWIFT-XRT-PHOTON > extract spectrum
extractor v4.47      1 Dec 2004
 Getting FITS WCS Keywords
 Doing file: /local/swift/sw00050300005xpcw4po_cl.evt
100% completed
```

```
         Total      Good    Bad: Region      Time     Phase     Grade     Cut
          4772        26           4746         0         0         0         0
===============================================================================
   Grand Total      Good   Bad:  Region      Time     Phase     Grade    Cut
         4772        26           4746         0         0         0         0
   in  12040.     seconds
 Spectrum         has       26 counts for 2.1595E-03 counts/sec
 ... written the PHA data Extension
xsel:SWIFT-XRT-PHOTON > save spectrum back.pha
Wrote spectrum to back.pha
xsel:SWIFT-XRT-PHOTON > clear region
===============================================================================
```

6.4.2 Extracting spectra for Photodiode mode using an intensity filter

The example below is the XSELECT log of the commands used to generate an intensity-selected spectrum for the Photodiode Mode. For this mode, there is no region selection applied because of its lack of positional information. The sequence used is from ground calibration data where two different intensity levels were included.

- Read the Photodiode mode event file
- Change setting to the new mode
- Set a grade selection to use only events with grade 0
- Extract and plot the lightcurve of all the data with 1 second binning

```
===========================================================================================
> xselect

                         **  XSELECT V2.2a  **

> Enter session name >[xsel]
xsel:ASCA > read events sw00073701002xlrb1po_cl.evt
> Enter the Event file dir >[./]
Got new mission: SWIFT
> Reset the mission ? >[yes]

 Notes: XSELECT set up for      SWIFT
 Time keyword is TIME       in units of s
 Default timing binsize =   1.0000

Setting...
 Image  keywords   = DETX       DETY        with binning =    1
 WMAP   keywords   = DETX       DETY        with binning =    1
 Energy keyword   = PI                     with binning =    1

Getting Min and Max for Energy Column...
Got min and max for PI:     0   1023
```

```
Got the minimum time resolution of the read data:  0.14000E-03
MJDREF =  5.1910000742870E+04 with TIMESYS = TT
 Number of files read in:  1

******************** Observation Catalogue ********************

Data Directory is: /local/swift
HK Directory is: /local/swift


        OBJECT      OBS_ID      DATE-OBS    DATAMODE
      1 XRT Ground  00073701002 2001-01-14T LOWRATE

xsel:SWIFT-XRT-LOWRATE > filter grade 0
xsel:SWIFT-XRT-LOWRATE > extract curve
extractor v4.48      9 Dec 2004
 Getting FITS WCS Keywords
 Doing file: /local/swift/sw00073701002xlrb1po_cl.evt
100% completed
          Total      Good    Bad: Region      Time     Phase     Grade     Cut
         120660     82643             0          0         0     38017         0
===============================================================================
   Grand Total      Good   Bad:  Region      Time     Phase     Grade    Cut
        120660     82643             0          0         0     38017         0
   in  165.24     seconds
 Fits light curve has    82643 counts for  500.1     counts/sec
 Thresholding removed more than half the bins
 Try exposure=0.0 on the extract command in xselect
 or lcthresh=0.0 if running extractor stand-alone
xsel:SWIFT-XRT-LOWRATE > plot curve
PLT> r y 0 2000
PLT> p
PLT> quit
==================================================================================================
```

- Set the intensity filter
- Extract the spectrum
- Save the spectrum to a file
- Exit `XSELECT`

```
==================================================================================================

xsel:SWIFT-XRT-LOWRATE > filter intensity 300-600
Making GTI to implement intensity selection with Boolean expression:
 * RATE .GE.    300.0     .AND. RATE .LE.    600.0
PREFR keyword not found, using prefr = 0.5
```

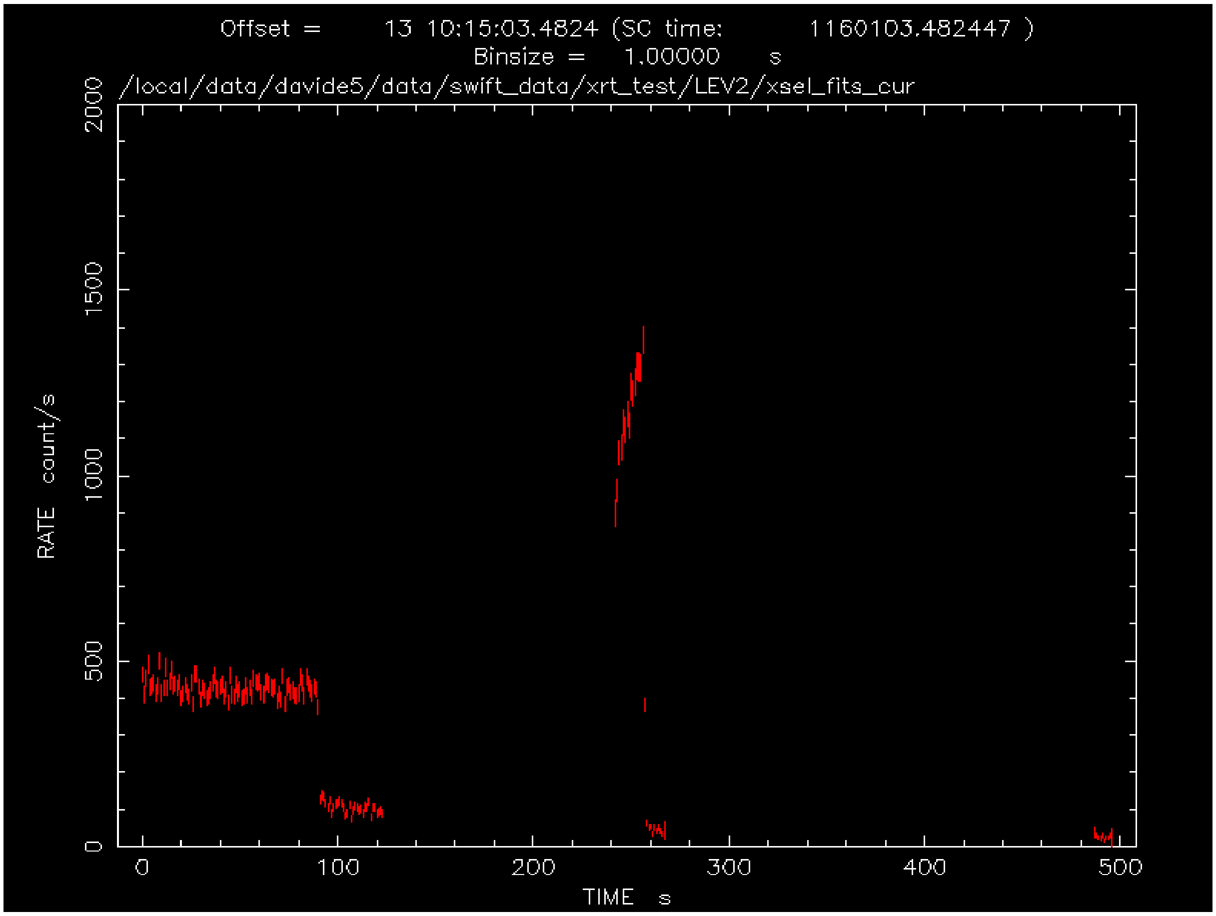

Figure 6.2: Curve Plot from which the Intensity values are selected

```
POSTFR keyword not found, using postfr = 0.5
xsel:SWIFT-XRT-LOWRATE > extract spectrum
extractor v4.48      9 Dec 2004
 Getting FITS WCS Keywords
 Doing file: /local/swift/sw00073701002xlrb1po_cl.evt
100% completed
          Total      Good    Bad: Region      Time      Phase      Grade      Cut
         120660     29108                0     83502          0       8050          0
===============================================================================
   Grand Total      Good    Bad:  Region      Time      Phase      Grade     Cut
        120660     29108                0     83502          0       8050          0
   in  86.500     seconds
 Spectrum         has     29108 counts for  336.5     counts/sec
 ... written the PHA data Extension
xsel:SWIFT-XRT-LOWRATE > save spectrum source_lr.pha
Wrote spectrum to source_lr.pha
xsel:SWIFT-XRT-LOWRATE > quit
> Save this session? >[yes] no
================================================================================================
```

6.5 Further analysis on the science products

The spectra, lightcurves and images can be used as input to XSPEC, XRONOS, and XIMAGE or any other software analysis software packages that accept the `XSELECT` outputs. After having creating the source and background spectra, spectral analysis packages (for example XSPEC) require response matrices and ancillary response files (arfs). For the XRT, the response matrices, for a specific set of grades, are available in CALDB (see also chapter 8). CALDB also contains a set of standards arf files derived for on-axis position and standard extraction radius, however arf at arbitrary detector positions and different extraction radius can be derived using the task `xrtmkarf`. For example to create an arf for the spectrum extracted with the Photodiode mode, xrtmkarf can be run as :

```
xrtmkarf phafile=source_lr.pha srcx=300 srcy=300 outfile=source_lr.arf psfflag=no
```

where the source position has been entered in detector coordinates and here were set to the detector center. CALDB needs to be initialized since the task uses many of the calibration file.

One of the common usage of the XRT images is to derive the position of the GRB and its error. This is accomplished with the specific XRT tasks `xrtcentroid` which uses `XIMAGE` to derive the centroid and calibration information stored in CALDB to derive the error. In interactive mode this task can be run as in the following example:

```
xrtcentroid infile=sw<obs_id>xpcwnpocl.evt outfile=DEFAULT outdir=./
calcpos=yes interactive=yes
```

which accepts as input an event file or an image and outputs an ASCII file containing the position and its error.

Chapter 7

XRT TDRSS messages

7.1 Introduction

For each new trigger detected by the BAT, the three instruments, the Figure of Merit algorithm, or the spacecraft can send messages on-ground via TDRSS, which are then broadcast by the GCN reporting on the first look at the new source. For the XRT, these TDRSS messages will be generated only if the satellite has slewed to the new position. The XRT TDRSS messages contain the results of on-board processing of the data acquired during the very first look at the new position. The data on which these results are based are later relayed via Malindi and included in the first observation at this new position.

7.2 The messages

The XRT messages containing the on-board processing results are :

- A position message reporting the XRT position of the new source as calculated on board.
- A postage stamp image centered on the XRT position of the new source.
- A centroid error-message reporting the reason why the source position could not be calculated on board.
- A spectrum in Photodiode mode and one in Windowed timing mode.
- A lightcurve.

The messages are formatted in FITS using either the primary header or a binary table extension. The information from the position and the centroid error messages is provided as keywords in a primary header of a FITS file with an empty array. The postage stamp image is stored in the primary header and the spectra and the lightcurve instead are in binary table extensions. The filenames of the XRT messages are :

```
XRT centroid        sw[obs_id]msxce.fits
XRT image           sw[obs_id]msxim_rw.img
XRT spectrum 1      sw[obs_id]msxlr_rw.pha
XRT spectrum 2      sw[obs_id]msxwt_rw.pha
```

```
 XRT no centroid     sw[obs_id]msxno.fits
 XRT lightcurve      sw[obs_id]msx.lc
```

On the ground the postage stamp image and the spectra are processed to produce files suitable for image and spectral analysis. The task *xrttdrss* applies the necessary transformations and produces three additional files that are also archived.

xrttdrss can process the postage stamp image and spectra individually or simultaneously, and it is run at the GCN before these messages are broadcast. For example, if both the postage stamp and the spectra are available, the command is :

```
xrttdss imagefile=file1 spec1file=file2 spec2file=file3
```

or

```
xrttdss imagefile=file1 spec1file=NONE  spec2file=NONE
```

if instead only the postage stamp image is available.

7.3 Position, postage stamp and centroid error

If the on-board XRT centroid was successful, the XRT sents out the position message. This includes the position of the source (J2000), the total rate observed, corresponding to the total charge (DN) per second, and the sqrt of the charge. These values are stored in the following keywords :

```
XRA_OBJ                        / [deg] XRT RA location of GRB or Object
XDEC_OBJ                       / [deg] XRT DEC location of GRB of Object
RATEDN                         / [adu/s] DN charge in standard window
SQRTDN                         / Sqrt of the DN
```

The RATEDN corresponds to the total charge collected in the window (sub-image) used to calculate the centroid.

The postage stamp image calculated on-board is obtained using the Image mode data (0.3-10 keV) and it is sent only if the centroid could be determined on-board. It is a 51x51 pixel image (2'x 2') in RAW coordinates (as defined in Chapter 3) and the bias has not been subtracted. On-ground the bias is subtracted, the image is transformed in a 'looking up' orientation, and the FITS world coordinate system keywords are written in the header. The resulting image is rotated with respect to the celestial north by the roll angle. The ground processing also calculates the error in the centroid, the total charge after bias subtraction, its rate, and the flux in erg/cm^2/s using a standard conversion factor. These values are stored in the following keywords :

```
XRTBIAS =                   / Bias calculated on ground [always applied]
ERRCTRD =                   / error on the XRT centroid position
SRCFLUX =                   / [erg/cm**2/s] source flux (0.3-10 keV)
TOTALDN =                   / Total DN value after bias subtraction
RATEDN  =                   / DN rate value (DN/EXPOSURE)
CONVFACT=                   / Conversion factor DN/s to 0.3-10keV flux
```

The TOTALDN is calculated using all 51x51 pixels in the image.

If the XRT cannot determinate a centroid position for the new discovered source, it sends down a centroid error message. This includes several diagnostics and an error flag reporting the reasons why the centroid could not be determined. The error flag is stored in the keyword ERRFLAG and it is set to one of the following values:

```
COMMENT ERRFLAG 1=no source found in the image
COMMENT         2=algorithm did not converge
COMMENT         3=standard deviation too large
COMMENT         0xFFFF= general error
```

7.4 Spectra

After the postage stamp image, the XRT sends down spectra obtained from the timing modes (Photodiode and Windowed Timing mode). The number of spectra depends on the source count rate. Each spectrum contains a 1024 bin histogram in PHA (not PI) binned by a factor of 4 compared to the original PHA array. The file format is compatible with the XSPEC format and contains the following columns:

TTYPE	TFORM	TUNIT	TLMIN	TLMAX	Description
CHANNEL	1I	chan	0	1023	Channel number
COUNTS	1J	count	-	-	Total count in the channel

The spectra are calculated without applying on-board event recognition, and therefore all pixels are included. In Photodiode mode, the spectra is accumulated using the entire field of view, including the corner Fe-55 calibration sources. Therefore, the spectra include the Mn K-alpha and K-beta emission lines at 5.9 and 6.4 keV. In Windowed Timing mode, the spectra includes all pixels in the 1-D window. The standard window setting is of 200 pixels, therefore the corner calibration sources are not included. The on-board software can send these spectra in different ways: the PD spectrum can have the bias subtracted (default) or not, and the WT spectrum can include also all the counts collected in PD mode or not. The keyword BIASLVL in the header of the PD spectrum reports the bias value, if it has been subtracted on-board, otherwise it is set to 0. The keyword DATAMODE in the header of the WT spectrum is set to MIXED if the counts are from both the WT and PD mode data, otherwise it is set to WINDOWED if the counts are from only from the WT mode data (default setting).

The task *xrttdrss* processes the two spectra. The output spectral files can be used for approximate spectral analysis with XSPEC using the appropriate PHA response matrix in PHA available for the TDRSS spectra (see Chapter 8). The spectra are not background-subtracted and the PD spectrum contains the line energy from the corner calibration sources. The calibration database contains ready-made background spectra for the PD and WT mode in PHA that can be used in XSPEC. These spectra give a qualitative view of the source spectral characteristics, but it is recommended that users derive the source spectra from the science data available through Malindi for any detailed quantitative analysis.

7.5 Lightcurve

The last XRT message is a lightcurve. The lightcurve contains data from all event modes, where each bin contains the observed count rate in one frame of data. The bin integration time therefore varies across the lightcurve and is set to the frame exposure for the different modes. These are 8.2623, 1.0675 and 2.5073 seconds for the PD, WT and PC, respectively. The lightcurve contains a maximum of 100 bins and its length in time depends on the scheduled mode. The file format is a binary table extension with the following columns:

TTYPE	TFORM	TUNIT	Description
TIME	1D	s	Time associated with the bin
RATE	1E	count/s	Total rate in the bin
TIMEDEL	1D	s	Integration of the bin
XRTMODE	1I		Numeric value indicating the XRT data mode

The TIME column records the bin time at the end of the integration. The values in the column RATE are accumulated on-board without accounting for event reconstruction or any other correction and should only be used to work out the qualitative behavior of the time decay of the source intensity.

Chapter 8

Calibration Files

8.1 Introduction

The software interfaces with the calibration information via the Calibration Database (CALDB). CALDB consists of a collection of files, each dedicated to a specific aspect of the calibration, organized in a specific directory structure. Software retrieves the CALDB files by querying an *index* file, that contains records of the calibration file and their validity.

The CALDB files for the XRT can be divided in three categories:

- Files used by the software to calibrate events, for example, to calculate the PI values, or to record standard screening criteria.
- Files used in the analysis of the extracted products, for example, the response matrices.
- Files not used directly by the software containing instrument characteristics and included in the calibration database for archival purposes.

The following sections list the XRT CALDB files that are directly used by the XRT software tasks and in the analysis of the extracted products. The complete list of XRT calibration files and their format are described in a separate document, "Description of the XRT Calibration Files", available from

$http://heasarc.gsfc.nasa.gov/docs/caldb/swift/.$

The calibration files are named according to the following convention:

swx[datatype][date]v[ver].[ext]

where [datatype] provides an identifier for the calibration data, [date] indicates the first date of validity of the file and [ver] is the version number for that file. At any time during the mission the calibration files are available from the above web address.

8.2 Calibration files listing

The tables in this section provide a quick reference for the calibration files required by the software tasks described in 4 and by the software used to analyse the high level data products (spectra and images).

Each table contain four columns. The first lists the name of the software task; the second the XRT mode for which that task is applicable; the third gives a brief description of the calibration files used by the task and the last column is the [datatype] in the filename of the calibration file and/or the file extension ([ext]) if different from '.fits'.

8.2.1 Calibration Files for the XRT Level 1 and Level 2 software

This list includes the calibration files used to produce the XRT Level 1 and Level 2 event files and directly used in the software described in the 4. Their names can be constructed using the [datatype]. For example the event grade calibration file defined pre-launch and valid also during the operation is is named *swxgrade20010101v002.fits*.

Software Task	Mode	Calibration File Description	[datatype]/[ext]
coordinator	PC	Telescope definition	/.teldef
xrtflagpix	PC WT	Bad pixel lists	badpix,onboardbp
		Calibration source regions	region
xrtpcgrade	PC	Event Grade for PC	pcgrade
xrtcalpi	PC PU WT	Instrument Gain	pcgain, pdgain, wtgain
xrthkproc	PD WT	Telescope definition	/.teldef
xrttimetag	PD WT	Telescope definition	/.teldef
xrtpdcorr	PD	Bias for the PD mode	pdbias
xrtevtrec	PD WT	Event Grade for PD and WT	pdgrade, wtgrade
xrtfilter	All	Configuration for prefilter	preconf
		Configuration for makefilter	mkfconf
xrtscreen	All	Standard HK values	hkrange
		Standard Event screening	evtrange
xrtimage	IM	Bad pixel	badpix,onboardbp
		Calibration source region	region
		Bias for IM mode	imbias
swiftxform	IM	Telescope definition	/.teldef
xrttam	All	Reference TAM Position	tamref

Table 8.1: Level 1 & 2 Calibration Files

8.2.2 Calibration Files used in the analysis software for high level data products

This list includes the calibration files required to calculate an error on a position obtained from the PC and IM mode data and to create an ancillary response file (ARF), which is than used for spectral analysis.

CALDB also includes a standard ARF generated for an on-axis position with a standard extraction radius for the Photon Counting, and Windowed Timing modes and considering the entire detector for the Photodiode mode (see below).

Software Task	Mode	Calibration File Description	[datatype]/[ext]
xrtmkarf	PC PU WT	Filter transmission	ftrans
		Vignetting	vign
		PSF	psf
		Effective area	effarea
		Response Matrix	(see section on rmf)
		Empirical arf	(see section on rmf)
xrtcentroid	PC IM	Position error	posserr

Table 8.2: Level 1 Photodiode Modes Fits File Events Table Columns

8.2.3 Response matrices and Standard ARF

The XRT response matrices that are available in CALDB are prepared for a standard events grade setting. There are two sets of responses. One set is suitable for spectra extracted in PI using the events data, the other is for spectra accumulated on-board and sent on the ground via the TDRSS messages.

The table 8.3 provides a quick reference of the responses which are appropriate for spectra extracted in PI from the event data, where the first and second columns list the applicable range of grades and the XRT data mode. The last column gives the [datatype] for the filename.

Grade	Mode	[datatype]/[ext]
0:0	PC PD WT	pc0_, pd0_, wt0_,
0:2	PD WT	pd0to2_,wt0to2_
0:4	PC	pc0to4_
0:5	PD	pd0to5_
0:12	PC	pc0to12_

Table 8.3: Response matrices

For example the response matrices valid for grade 0 and grade 0-2 for the Photon Counting mode are named swxpc0_20010101v006.rmf and swxpc0to2_20010101v006.rmf respectively where the date and version number will change during the mission.

The second set includes two additional responses for spectra accumulated on-board with the PD and the WT modes in PHA and sent via TDRSS. These responses should not be used with the spectra extracted in PI from the event modes. Their [datatype] is 'pdmspha' and 'wtmspha' for the PD and WT modes respectively. Their filenames is therefore swxpdmspha_20010101v006.rmf and swxwtmspha_20010101v006.rmf for the PD and WT modes respectively where the date and version number will change during the mission. NOTE: User should be aware that the spectra send via TDRSS should be used with caution since they are binned on-board considering all pixels. Although there are response matrices available these were released to facilitate the mapping into energy and any spectral analysis should be considered preliminary and any finding should be confirmed by extracting the spectra in PI from the calibrated event file before publication.

Currently the calibration is still on-going and there are two different methods to calculate the arfs using *xrtmkarf*: *physical* where the basic calibration files such as filter transmission, effective area, PSF and vignetting are used to generate the arf (the *xrtmkarf* parameter 'inarffile' should be set to NONE) and *empirical* where already made ARFs stored in CALDB are corrected for the PSF and Vignetting. These already made ARFs were obtained by adjusting the ARF to fit the Crab spectrum by using the latest response matrices, therefore they are also grade dependent and use the same naming convention adopted for the rmf. CALDB includes the standard on-axis ARFs, generated as above, for the Photon Counting, Windowed Timing and Photodiode modes.

8.2.4 Standard background spectra in PHA

Pre-launch background spectra derived in PHA channels are available from CALDB. These are 1024 channel spectra binned by a factor of four compared to the normal PHA range and include only the detector background counts. The pre-launch background spectra are available for the Photodiode, Windowed Timing and Photon Counting mode data. Their filenames are swxlrbkg-pha[date]v[ver].pha, swxwtbkgpha[date]v[ver].pha and swxpcbkgpha[date]v[ver].pha for the three modes respectively.

They are suitable to subtract the detector background in spectra extracted using the PHA column in the event file and rebinned by four. They are not suitable if the source spectra is extracted using the column PI in the event file.

The PD and WT mode spectra sent down via TDRSS are not background subtracted and these background files can used within XSPEC to account for the detector background particularly important for the PD spectra since includes the emission from the corner calibration sources.

Appendix A

FITS file structure

A.1 Photodiode Modes FITS File Format

A.1.1 Level 1 or the *uf* File Format

The Low rate and the Pileup photodiode modes have both the same format. The file structure is :

HDU	Type	EXTNAME	Dim(col)	Description
0	PRIMARY	-	0	Primary Header
1	BINTABLE	EVENTS	37(9)	Events Extension
2	BINTABLE	GTI	16(2)	GTI Extension

Table A.1: Level 1 Photodiode Modes Fits File structure

The columns in the EVENTS extension are :

TTYPE	TFORM	TUNIT	TLMIN	TLMAX	TZERO	Description
TIME	1D	s	-	-	-	Calculated Photon Arrival Time
CCDFrame	1J	-	-	-	2147483648	CCD Frame Number
OFFSET	1J	pixel	-	-	-	Offset counter
DETX	1I	pixel	1	600	-	Det X position of target
DETY	1I	pixel	1	600	-	Det Y position of target
PHA	1J	chan	0	4095	-	DN value Bias subtracted
Amp	1B	-	-	-	-	Amplifier Number (1 or 2)
ROTIME	1D	s	-	-	-	Frame ReadOut Time

Table A.2: Level 1 Photodiode Modes Fits File Events Table Columns

A.1.2 Level 1a or the *ufre* File Format

The file structure is :

HDU	Type	EXTNAME	Dim(col)	Description
0	PRIMARY	-	0	Primary Header
1	BINTABLE	EVENTS	49(12)	Events Extension
2	BINTABLE	GTI	16(2)	GTI Extension
3	BINTABLE	BIAS	12(2)	BIAS Extension

Table A.3: Level 1a Photodiode Modes Fits File structure

The columns in the EVENTS extension are :

TTYPE	TFORM	TUNIT	TLMIN	TLMAX	TZERO	Description
TIME	1D	s	-	-	-	Calculated Photon Arrival Time
CCDFrame	1J	-	-	-	2147483648	CCD Frame Number
OFFSET	1J	pixel	-	-		Offset counter
DETX	1I	pixel	1	600	-	Det X position of target
DETY	1I	pixel	1	600	-	Det Y position of target
PHA	1J	chan	0	4095	-	Reconstructed PHA
Amp	1B	-	-	-	-	Amplifier Number (1 or 2)
PHAS	7I	chan	0	4095	-	The DN value of the pixel and the DN value of the 7x1 centered on this pixel
PixsAbove	1I	-	-	-	-	Number of pixels above the event threshold used to reconstruct PHA value
GRADE	1I	-	0	15	-	PHAS derived photon pattern
PI	1J	chan	0	1023	-	Pulse Invariant value
STATUS	16X	-	-	-	-	Quality event flag
EVTPHA	1J	chan	0	4095	-	Telemetred DN value
ROTIME	1D	s	-	-	-	Readout Time

Table A.4: Level 1a Photodiode Modes Fits File Events Table Columns

The standard processing removes the EVTPHA column and does not add the PHAS and PixsAbove optional columns, because these columns are not useful for Stage 2.

TTYPE	TFORM	TUNIT	TLMIN	TLMAX	TZERO	Description
CCDFrame	1J	-	-	-	2147483648	CCD Frame Number
BIAS	1D	s	-	-	-	Bias value used for this frame

Table A.5: Level 1a Photodiode Modes Fits File Bias Table Columns

The standard processing adds the BIAS extension only if the bias is subtracted on ground.

A.1.3 Level 2 or *cl* File Format

The file structure is :

HDU	Type	EXTNAME	Dimensions(columns)	Description
0	PRIMARY	-	0	Primary Header
1	BINTABLE	EVENTS	24(7)	Events Extension
2	BINTABLE	GTI	16(2)	GTI Extension

Table A.6: Level 2 Photodiode Modes Fits File structure

The columns in the EVENTS extension are :

TTYPE	TFORM	TUNIT	TLMIN	TLMAX	TZERO	Description
TIME	1D	s	-	-	-	Calculated Photon Arrival Time
DETX	1I	pixel	1	600	-	Det X position of target
DETY	1I	pixel	1	600	-	Det Y position of target
PHA	1J	chan	0	4095	-	Reconstructed PHA
GRADE	1I	-	0	15	-	PHAS derived photon pattern
PI	1J	chan	0	1023	-	Pulse Invariant value
STATUS	16X	-	-	-	-	Quality event flag

Table A.7: Level 2 Photodiode Modes Fits File Events Table Columns

A.2 Windowed Timing Mode Fits File Format

A.2.1 Level1 or the *uf* File Format

The file structure is :

HDU	Type	EXTNAME	Dim(col)	Description
0	PRIMARY	-	0	Primary Header
1	BINTABLE	EVENTS	39(12)	Events Extension
2	BINTABLE	GTI	16(2)	GTI Extension
3	BINTABLE	BADPIX	12(6)	Bad pixels Extension

Table A.8: Level 1 Windowed Timing Mode Fits File structure

The columns in the EVENTS extension are :

TTYPE	TFORM	TUNIT	TLMIN	TLMAX	TZERO	Description
TIME	1D	s	-	-	-	Calculated Photon Arrival Time
CCDFrame	1J	-	-	-	2147483648	CCD Frame Number
X	1I	pixel	1	600	-	X position of target
Y	1I	pixel	1	600	-	Y position of target
RAWX	1I	pixel	0	599	-	Raw X position of photon
RAWY	1I	pixel	0	599	-	Offset counter
DETX	1I	pixel	1	600	-	Det X position of photon
DETY	1I	pixel	1	600	-	Det Y position of target
PHA	1J	chan	0	4095	-	DN value of this pixel
Amp	1B	-	-	-	-	Amplifier Number (1 or 2)
STATUS	16X	-	-	-	-	Quality event flag
ROTIME	1D	s	-	-	-	Frame ReadOut Time

Table A.9: Level 1 Windowed Timing Mode Fits File Events Table Columns

A.2.2 Level 1a or *ufre* File Format

The file structure is :

HDU	Type	EXTNAME	Dim(col)	Description
0	PRIMARY	-	0	Primary Header
1	BINTABLE	EVENTS	53(15)	Events Extension
2	BINTABLE	GTI	16(2)	GTI Extension
3	BINTABLE	BADPIX	12(6)	Bad pixels Extension

Table A.10: Level 1a Windowed Timing Mode Fits File structure

The columns in the EVENTS extension are :

TTYPE	TFORM	TUNIT	TLMIN	TLMAX	TZERO	Description
TIME	1D	s	-	-	-	Calculated Photon Arrival Time
CCDFrame	1J	-	-	-	2147483648	CCD Frame Number
X	1I	pixel	1	600	-	X position of target
Y	1I	pixel	1	600	-	Y position of target
RAWX	1I	pixel	0	599	-	Raw X position of photon
RAWY	1I	pixel	0	599	-	Offset counter
DETX	1I	pixel	1	600	-	Det X position of photon
DETY	1I	pixel	1	600	-	Det Y position of target
PHA	1J	chan	0	4095	-	Reconstructed PHA
Amp	1B	-	-	-	-	Amplifier Number (1 or 2)
STATUS	16X	-	-	-	-	Quality event flag
PHAS	7I	chan	0	4095	-	The DN value of the pixel and the DN value of the 7x1 centered on this pixel
PixsAbove	1I	-	-	-	-	Number of pixels above the event threshold used to reconstruct PHA value
GRADE	1I	-	0	15	-	PHAS derived photon pattern
PI	1J	chan	0	1023	-	Pulse Invariant value
EVTPHA	1J	chan	0	4095	-	Telemetred DN value
ROTIME	1D	s	-	-	-	Readout Time

Table A.11: Level 1a Windowed Timing Mode Fits File Events Table Columns

The standard processing removes the EVTPHA column and does not add the PHAS and PixsAbove optional columns, because these columns are not useful for Stage 2.

A.2.3 Level 2 or the *cl* File Format

The file structure is :

The columns in the EVENTS extension are :

HDU	Type	EXTNAME	Dim(col)	Description
0	PRIMARY	-	0	Primary Header
1	BINTABLE	EVENTS	53(15)	Events Extension
2	BINTABLE	GTI	16(2)	GTI Extension
3	BINTABLE	BADPIX	12(6)	Bad pixels Extension

Table A.12: Level 2 Windowed Timing Mode Fits File structure

TTYPE	TFORM	TUNIT	TLMIN	TLMAX	TZERO	Description
TIME	1D	s	-	-	-	Calculated Photon Arrival Time
X	1I	pixel	1	600	-	X position of target
Y	1I	pixel	1	600	-	Y position of target
RAWX	1I	pixel	0	599	-	Raw X position of photon
RAWY	1I	pixel	0	599	-	Offset counter
DETX	1I	pixel	1	600	-	Det X position of photon
DETY	1I	pixel	1	600	-	Det Y position of target
PHA	1J	chan	0	4095	-	Reconstructed PHA
STATUS	16X	-	-	-	-	Quality event flag
GRADE	1I	-	0	15	-	PHAS derived photon pattern
PI	1J	chan	0	1023	-	Pulse Invariant value

Table A.13: Level 2 Windowed Timing Mode Fits File Events Table Columns

A.3 Photon Counting mode

The file structure is :

A.3.1 Level 1 or the *uf* File Format

HDU	Type	EXTNAME	Dim(col)	Description
0	PRIMARY	-	0	Primary Header
1	BINTABLE	EVENTS	57(15)	Events Extension
2	BINTABLE	GTI	16(2)	GTI Extension
3	BINTABLE	BADPIX	12(6)	BADPIX Extension

Table A.14: Level 1 Photon Counting Mode Fits File structure

The columns in the EVENTS extension are :

TTYPE	TFORM	TUNIT	TLMIN	TLMAX	TZERO	Description
TIME	1D	s	-	-	-	Photon Arrival Time
CCDFrame	1J	-	-	-	2147483648	CCD Frame Number
X	1I	pixel	1	600	-	X pos. on sky
Y	1I	pixel	1	600	-	Y pos. on sky
RAWX	1I	pixel	0	599	-	Raw X pos. of ph.
RAWY	1I	pixel	0	599	-	Raw Y pos. of ph.
DETX	1I	pixel	1	600	-	Det X pos. of ph.
DETY	1I	pixel	1	600	-	Det Y pos. of ph.
PHAS	9I	chan	0	4095	-	The DN value of the pixel and the DN value of the 3x3 centered on this pixel
Amp	1B	-	-	-	-	Amplifier Number (1 or 2)
PHA	1J	chan	0	4095	-	Reconstructed PHA value
PI	1J	chan	0	1023	-	Pulse Invariant value
GRADE	1I	-	0	32	-	PHAS derived photon pattern
PixsAbove	1I	-	-	-	-	Number of pixels above the event thr used to reconstruct PHA value
STATUS	16X	-	-	-	-	Quality event flag

Table A.15: Level 1 Photon Counting Mode Fits File Events Table Columns

A.3.2 Level 2 or the *cl* File Format

The file structure is :

HDU	Type	EXTNAME	Dim(col)	Description
0	PRIMARY	-	0	Primary Header
1	BINTABLE	EVENTS	32(11)	Events Extension
2	BINTABLE	GTI	16(2)	GTI Extension
3	BINTABLE	BADPIX	12(6)	BADPIX Extension

Table A.16: Level 2 Photon Counting Mode Fits File structure

The columns in the EVENTS extension are :

TTYPE	TFORM	TUNIT	TLMIN	TLMAX	TZERO	Description
TIME	1D	s	-	-	-	Photon Arrival Time
X	1I	pixel	1	600	-	X pos. on sky
Y	1I	pixel	1	600	-	Y pos. on sky
RAWX	1I	pixel	0	599	-	Raw X pos. of ph.
RAWY	1I	pixel	0	599	-	Raw Y pos. of ph.
DETX	1I	pixel	1	600	-	Det X pos. of ph.
DETY	1I	pixel	1	600	-	Det Y pos. of ph.
PHA	1J	chan	0	4095	-	Reconstructed PHA value
PI	1J	chan	0	1023	-	Pulse Invariant value
GRADE	1I	-	0	32	-	PHAS derived photon pattern
STATUS	16X	-	-	-	-	Quality event flag

Table A.17: Level 2 Photon Counting Mode Fits File Events Table Columns

A.4 GTI and Bad Pixel table FITS Format

All event files have a GTI extension. The format is :

TTYPE	TFORM	TUNIT	TLMIN	TLMAX	TZERO	Description
START	1D	s	-	-	-	GTI Start Time
STOP	1D	s	-	-	-	GTI Stop Time

Table A.18: Generic GTI Table

The Windowed Timing and Photon Counting modes have a Bad pixel extension. The format is :

TTYPE	TFORM	TUNIT	TLMIN	TLMAX	TZERO	Description
RAWX	1I	pixel	0	599	-	Raw X position of bad pixel
RAWY	1I	pixel	0	599	-	Raw X position of bad pixel
Amp	1B	-	-	-	-	Amplifier Number (1 or 2)
TYPE	1I	pixel	1	600	-	1=Point 2=Column
YEXTENT	1I	pixel	1	600	-	
BADFLAG	16X	-	-	-	-	Bad pixels flag

A.5 Short and Long Image Fits File Format

The format for the Short and Long images is identical. The Level 1 where data are in RAW coordinates differs from the Level 2 in sky coordinates by the dimension of the array

A.5.1 Level 1

HDU	Type	EXTNAME	Dim(col)	Description
0	PRIMARY	-	0	Primary header
1	IMAGE	SHTxxxxxxxxxI	600 x 600	Image extension

Table A.19: Level 1 Short and Long Image Fits File structure

A.5.2 Level 2

HDU	Type	EXTNAME	Dim(col)	Description
0	PRIMARY	-	0	Primary header
1	IMAGE	SHTxxxxxxxxxI	1000 x 1000	Image extension

Table A.20: Level 2 SKY Image Mode Fits File structure

A.6 *hd* Housekeeping File

The file structure of the *hd* HK file is :

HDU	Type	EXTNAME	Dim(col)	Description
0	PRIMARY	-	0	Primary Header
1	BINTABLE	FRAME	516(93)	

Table A.21: Housekeeping Fits File structure

The columns in the FRAME extension are :

TTYPE	TFORM	TUNIT	TLMIN	TLMA X	TZERO	Description
FrameHID	1J					Frame Identifier
CCDFrame	1J					CCD Frame Number
TARGET	1J					Target Number
ObsNum	1I					Observation Number
RA	1E	deg	0	360		RA of this frame
Dec	1E	deg	-90	90		Dec of this frame
Roll	1E	deg				Roll of this frame
Settled	1X					Observation mode
In10Arcm	1X					ACS Flag
InSAA	1X					SAA Flag
InSafeM	1X					XRT Status
XRTAuto	1X					XRT auto Flag
XRTManua	1X					XRT Manual Flag
XRTRed	1X					XRT Red Flag
XRTMode	1B		1	1		XRT Readout Mode
WaveID	1B					CCD Waveform use d
ErpCntRt	1E					ERP count rate
XPosTAM1	1E					X Position of TAM 1
YPosTAM1	1E					Y Position of TAM 1
XPosTAM2	1E					X Position of TAM 2
YPosTAM2	1E					Y Position of TAM 2
DNCCDTemp	1I					Frame CCD temperature
Vod1	1D	V				Output drain voltage for Amp 1
Vod2	1D	V				Output drain voltage for Amp 2
Vrd1	1D	V				Reference voltage for Amp 1
Vrd2	1D	V				Reference voltage for Amp 2
Vog1	1D	V				Output gate voltage for Amp 1
Vog2	1D	V				Output gate voltage for Amp 2
S1Rp1	1D	V				Serial register clock phase 1
S1Rp2	1D	V				Serial register clock phase 2
S1Rp3	1D	V				Serial register clock phase 3
R1pR	1D	V				Reset gate clock Amp 1
R2pR	1D	V				Reset gate clock Amp 2
S2Rp1	1D	V				Serial register clock phase 1
S2Rp2	1D	V				Serial register clock phase 2
S2Rp3	1D	V				Serial register clock phase 3
Vgr	1D	V				Guard ring bias voltage
Vsub	1D	V				Substrate bias voltage
Vbackjun	1D	V				Back junction bias voltage
Vid	1D	V				Input diode bias voltage
Ip1	1D	V				Image aria parallel clock phase 1
Ip2	1D	V				Image aria parallel clock phase 2
Ip3	1D	V				Image aria parallel clock phase 3

Table A.22: Housekeeping File FRAME Table Columns (1 part)

TTYPE	TFORM	TUNIT	TLMIN	TLMA X	TZERO	Description
Sp1	1D	V				store area parall clock ph1
Sp2	1D	V				store area parall clock ph2
Sp3	1D	V				store area parall clock ph3
CpIG	1D	V				Input gate clock
BaseLin1	1D	V				Basel volt signal chain A
BaseLin2	1D	V				Basel volt signal chain B
FSTS	1J	s				frame start time (sec)
FSTSub	1I	2*10**(-5)s	0	49999		frame start time (subsec)
FETS	1J	s				frame stop time (sec)
FETSub	1I	2*10**(-5)s	0	49999		frame stop time (subsec)
CCDExpos	1J	s				Nom. CCD frame time(s)
CCDExpSb	1I	2*10**(-5)s	0	49999		Nom. CCD frame time(subs)
EvtLLD	1I					Lower level discriminator
PixGtLLD	1J					Pixel greater than LLD
EvtULD	1I					Upper level discriminator
PixGtULD	1J					pixel greater ULD
SplitThr	1I					Th for splits in 3x3
OuterThr	1I					Th for vetos in 5x5
SnglePix	1I					N. of single pixel events
SngleSpl	1I					N. of singly-split events
ThreePix	1I					N. of three pixel events
FourPix	1I					N. of four pixel events
WinHalfW	1I					Half-width in pixels
WinHalfH	1I					Half-height in pixels
Amp	1B		1	2		Amplifier
telemRow	1I					N. of rows in telem
telemCol	1I					N. of columns in telem
nPixels	1J					N. of pixels following head
BaseLine	1J					Offset added to each DN
PixOverF	1J					N. of pixels overflows 4095
PixUnder	1J					N. of pixels underflows 0
TIME	1D	s				Frame start time
ENDTIME	1D	s				Frame stop time
BiasExpo	1I					Total n. of Bias Frame used
CCDTemp	1E					CCD temperature
LRHPixCt	1I					Total n. of pixels in header
LRBPixCt	1J					Tot n. of pix following head
LRSumBPx	1D					Sum of val of pix below LLD
LRSoSBPx	1D					Sum of Sq of pix below LLD
LRBiasPx	20I					DN of the last 20 pixels
LRBiasLv	1D					LRSumBPx/LRBPixCt
LRNoise	1D					LR calculated Noise
EVTLOST	1D					T. n. of the lost events
PDBIASTHR	1I					PD Bias Threshold
PDBIASLVL	1D					PD Bias Level

Table A.23: Housekeeping File FRAME Table Columns (2 part)

TTYPE	TFORM	TUNIT	TLMIN	TLMA X	TZERO	Description
LDPNUM	1J					Telemetry LDP Product Number
LDPPAGE	1J					Telemetry LDP Page Number
WTHPixCt	1I					Windowed Timing Bias PiXels
WTBiasPx	20I					Observation Number
HKTIME	1D					Housekeeping readout time

Table A.24: Housekeeping File FRAME Table Columns (3 part)

A.7 Filter File

The columns in the *mkf* file are :

TTYPE	TFORM	TUNIT	TLMIN	TLMAX	TZERO	Description
TIME	1D					Frame start time
ACS_SAA	1B					ACS reports in SAA
SAFEHOLD	1B					ACS reports in SAFE
SETTLED	1B					ACS reports settled on target
TEN_ARCMIN	1B					ACS reports within 10 arcmin of target
ANG_DIST	1E					Inst. point root mean sq. dev.
BR_EARTH	1E					Bright Earth Angles
COR_SAX	1E					Cut-off Rigidity
DEC	1E					Declination
ELV	1E					Elevation Angle
FOV_FLAG	1I					Field of View Flag
MCILWAIN_L	1E					
MOON_ANGLE	1E					Moon Angle
RA	1E					Right Ascension
RAM_ANGLE	1E					
ROLL	1E					
SAA	1I					South Atlantic Anomaly
SAA_TIME	1E					Time in SAA
SUNSHINE	1I					
SUN_ANGLE	1E					Angle with sun
Baselin1	1D	V				Basel volt for signal chain A
Baselin2	1D	V				Basel volt for signal chain B
CCDTemp	1E					CCD temperature
ENDTIME	1D	s				Frame stop time
PixGtULD	1J					pixel greater ULD
Vbackjun	1D	V				Back junction bias voltage
Vod1	1D	V				Output drain voltage for Amp 1
Vod2	1D	V				Output drain voltage for Amp 2
Vrd1	1D	V				Reference voltage for Amp 1
Vrd2	1D	V				Reference voltage for Amp 2
Vsub	1D	V				Substrate bias voltage

Table A.25: Filter File FILTER Table Columns

Appendix B

XRT SOFTWARE HELP

B.1 xrt tasks

The XRT software consists in the following tasks:

- xrtcalcpi - Update PI column in XRT event files (PC, WT and PD modes).
- xrtcentroid - Calculate centroid for the PC and IM mode
- xrtevtrec - Reconstruct events, calculate PHA and assign grade for the WT and PD modes.
- xrtfilter - Run 'prefilter' and 'makefilter' to create a filter file from HK data.
- xrtflagpix - Flag events for bad pixels and calibration source location.
- xrthkproc - Process XRT housekeeping header packets file.
- xrthotpix - Search for hot and flickering pixels for XRT Photon Counting mode.
- xrtimage - Subtract bias and clean bad pixels in Imaging Mode data.
- xrtmkarf - Generate an ARF file for an input RMF file.
- xrtpcgrade - Calculate the PHA values and assign event grades for PC data.
- xrtpdcorr - Check the bias and Subtract if necessary for data taken in PD mode.
- xrtproducts - Generate high level product data files from a cleaned event file.
- xrtscreen - Generate GTIs and use them together with other criteria to screen the data.
- xrttam - Calculate corrections to the attitude file using parameters from the TAM device.
- xrttdrss - Process XRT TDRSS messages.
- xrttimetag - Time tag events and calculate the DETX/DETY columns for the WT and PD modes.

In addition the script *xrtpipeline* run in sequence all the tasks for XRT data processing. The help for the individual tasks is provided below. With the software installed,the help can be viewed by using the command **'fhelp taskname'**.

B.1.1 xrtcalcpi

```
xrtcalcpi  infile outfile [parameter = <value> ...]
```

xrtcalcpi calculates the Pulse Invariant (PI) values in the Swift XRT event files, for data obtained with the Photon Counting (PC),Windowed Timing (WT) and Photodiode (PD) modes. The PI values are corrected for temporal changes in the gain and for positional gain variation (Charge Transfer Inefficiency, CTI). *xrtcalcpi* can be re-run on an event file if new calibration information are available. The PI column as well as the keyword XRTPI in the output file are updated by this task.

To calculate the PI values, *xrtcalcpi* uses the PHA values, stored in the PHA column of the input event file, the nominal gain and a set of coefficients describing the temperature and spatial dependence of the gain. The coefficient values have been evaluated from ground calibration data and are periodically updated based on the results of the flight calibration data analysis. The gain information is stored in the XRT calibration gain files which are included in CALDB. There is one similarly formatted gain calibration file for each of the modes. The nominal gain is in the header keyword **NOM_GAIN** and the coefficients that describe the temperature and the spatial dependence are stored in the columns GC0 - GC5 containing an array of three values one for each temperature. The gain correction applied to the event data is the result of a double interpolation of the coefficient values one in temperature and one in time, using the temperature and time closest to the temperature and epoch of the observation.

xrtcalcpi by default uses the XRT gain file, appropriate for each mode, stored in CALDB, but it can be replaced with a gain file input by the user (parameter *gainfile*). The user-provided gain file must have the same format as the XRT gain file in the Calibration Database. The algorithm describing the spatial dependence of gain is expressed in raw coordinates, RAWX and RAWY. For the timing modes (WT and PD) since the telemetry does not contain complete spatial information, *xrtcalcpi*, to account for the positional gain correction, assumes all events at the source location. For the WT mode, the RAWX coordinate of the event is known and the RAWY is obtained by transforming the DETY column value. For the PD mode the event detector coordinates are read from the DETX/DETY columns of the event file and transformed into RAWX/RAWY.

The nominal gain of the instrument is currently set to 10 eV per channel and is used by *xrtcalcpi*, if the parameter 'gainnom' is set to a negative value (default). Users can adjust the PI values via the parameter *offset* and *gainnom*. These values can be obtained with the XSPEC command 'gain' that provides a slope and a constant. The parameter *offset* should be set equal to the constant obtained by the XSPEC *gain* command and the parameter *gainnom* should be set equal to the slope obtained by the XSPEC 'gain' command multiplied by the default value of eV per channel.

Parameter

- infile [file name]
  Name of the input event FITS file. Unix-compressed files are allowed, except when the output file is set to NONE and the input file is overwritten.
- outfile [file name]
  Name of output event FITS file. The value 'NONE' will cause the input file to be overwritten.
- hdfile [file name]
  Name of input Housekeeping Header Packets FITS file.
- (gainfile = CALDB) [file name]
  Name of the gain file. If the parameters is set to CALDB, the file is read from the calibration database.
- (gainnom = -99.9) [real]
  Nominal gain value (eV/Channel). If 'gainnom' is negative (default), *xrtcalcpi* uses the value in the keyword **NOM_GAIN** of the CALDB gain calibration file currently set to 10 eV/channel.
- (offset = 0.0) [real]
  This parameter allows users to specify an offset (in keV) in the channel-energy relationship.
- (randomflag=yes) [boolean]
  If 'randomflag'=yes (default), the PHA values will be randomized.

- (seed = -1457) [integer]
  Random number generator seed, used to randomize the PHA values.
- (clobber=no) [boolean]
  If 'clobber'=yes and outfile=filename, the file with the same name will be overwritten if it exists.
- (history=yes) [boolean]
  If 'history=yes' the parameter values and other information are written in HISTORY keywords.
- (chatter = 2) [integer]
  Chatter Level (min=0, max=5)

B.1.2 xrtcentroid

```
xrtcentroid infile outfile outdir [parameter = < value >]
```

xrtcentroid calculates the source centroid an its associated error for the Swift XRT instrument. *xrtcentroid* accepts as input file data taken with the Swift XRT Photon Counting mode (event file),sky images taken with the Image mode and TDRSS postage stamp images. The centroid is estimated using the 'centroid' command of the XIMAGE package and it is calculated in a box region whose center and half-width can be specified with the parameters 'boxra', 'boxdec' and 'boxradius'. Alternatively, if the parameter 'interactive' set to 'yes', the box region can be defined interactively with the cursor.

The XIMAGE 'centroid' command calculates in the selected region the first guess for the centroid using two possible methods and then uses this best guess for the final calculation. The first guess for the centroid is obtained via a subsequent rebinning of the selected area until the location of maximum is the same as the last rebin (parameter 'hist'=no, default). The alternative method, selected by setting the parameter 'hist' to yes, determines the X and Y maximum from the distributions obtained by summing the counts in the pixels along the X and Y directions. The first best guess for the centroid is used as new center for the final centroid calculation. This is also achieved via two possible methods. The default method ('deriv'=no) re-evaluates the barycenter in boxes reduced by 80% each time from the original selected area. The alternative, selected by specifying the parameter 'deriv' to 'yes', uses the derivative of partial sums method.

The error circle on the position is derived by adding in quadrature the contributions of fours different components: i) statistical uncertainty on the centroid determination, which depends on the source total (time-integrated) intensity; ii) instrument residual misalignment; iii) spacecraft attitude reconstruction accuracy; iv) systematic error. The first was derived from ground calibration, the others are set to pre-launch nominal values. The four error components are recorded in a calibration file included in CALDB.

xrtcentroid allows also to calculate only the centroid error. In this case the user must provide in input the source total (time-integrated) intensity and its units ('COUNTS' for Photon Counting mode and 'DN' for Imaging mode) through the parameters 'totalint' and 'unit'. The centroid and error are written in an output file.

Parameters

- infile [file name]
  Name of the input event or image FITS file.
- outfile [file name]
  Name of the output file (ASCII).If DEFAULT the stem of the input file is used (option available only if parameter 'calcpos' is set to 'yes').
- outdir [directory name]
  Name of the output directory.

- posfile [file name]
  Name of the calibration file containing information on the centroid error components. If CALDB is input (default), the file included in CALDB is used.
- calcpos [boolean]
  If set to 'yes' calculates the centroid position and its error. If set to 'no' only the position error is calculated and the value of the parameters 'totalint' and 'unit' must be provided.
- totalint [real]
  Total (time-integrated) COUNTS (Photon Counting mode) or DN (Imaging mode) of the source. It is used to calculate the centroid position error if parameter 'calcpos' is set to 'no'.
- unit [string]
  String indicating the units for the 'totalint' parameter. Allowed units are 'COUNTS' for Photon Counting mode and 'DN' for Imaging mode
- interactive [boolean]
  If set to 'yes', use the cursor to define center (boxra boxdec) and size of box (boxradius).
- boxra [real]
  RA of the box center (degrees or hh mm ss.s). To be used if parameter 'interactive' is set to 'no'.
- boxdec [real]
  DEC of the box center (degrees or hh mm ss.s). To be used if parameter 'interactive' is set to 'no'.
- boxradius [real]
  Half-width of the box in arcmin.To be used if parameter 'interactive' is set to 'no'.
- (hist=no) [boolean]
  If set to 'yes' uses histogram method for the first preliminary centroid estimation.
- (deriv=no) [boolean]
  If set to 'yes' uses partial derivate method for the final centroid determination.
- (clobber = no)
  If set to 'yes' overwrites the output files with the same name if they exist.
- (chatter = 3) [integer]
  Verbosity Level from 0 to 5.
- (cleanup = yes) [boolean]
  If set to 'yes' deletes temporary files.

B.1.3 xrtevtrec

```
xrtevtrec infile outfile [parameter=<value>]
```

xrtevtrec processes the Swift XRT event files containing data taken with the Windowed Timing (WT) and Photodiode (PD) modes to reconstruct events, calculate their PHA values and assign a grade to each event.

The event reconstruction is carried out by searching for the local maximun in the neighborhood of a 7x1 pixel array. The PHA value of a 'reconstructed' event is obtained by summing up the PHA of the central pixel with that of the surrounding pixels above the split threshold and the result is written in the PHA column. The event grade is assigned using pre-defined patterns for the charge distribution and its value is stored in the GRADE column. A calibration file included in CALDB contains the charge distribution patterns of the different grades.

If the parameter *addcol* is set to 'yes', the columns PHAS, containing the 7x1 pixel array used for the event reconstruction, and PixsAbove, containing the number of pixels above split threshold, are added to the output file. All the other columns present in the events extension of the input file are copied to the output file. The original PHA value of each pixel is copied into a column named EVTPHA.

After the event reconstruction only a subset of pixels are recognized as the event. All pixels associated with a local maximum that have contributed to the reconstruction of one event have the GRADE and PHA columns assigned to NULL. These pixels are by default kept in the output file. However it is possible to exclude them from the output by setting the parameter 'delnull' to 'yes'.

xrtevtrec allows the user to set the event and split thresholds using the parameters 'event' and 'split'. In addition, for the Windowed Timing mode, it is possible to flag a reconstructed event for which the 7x1 pixel array contains a bad column. This is obtained by setting the parameter 'flagneigh' to yes and causes the STATUS column to be updated.

xrtevtrec adds the keyword XRTEVREC to the output file header set to 'T' to indicate that event reconstruction has been done. The values of the event threshold and split threshold used in the calculation are recorded in the keywords EVENTTHR and XRTSPLIT. The percentage of saturated events, the percentage of pixel not included in the reconstruction, and the percentage of pixels with PHA set to NULL are recorded in the keywords SATEVPER, NORECPER, NULLPER, respectively.
The Data Subspace convention (DS keywords) recording the event grade values present in the file (DSVALn, DSTYPn, DSFORMn) are also added to the output file.

Parameters

- infile [file name]
  Name of the input event FITS file. Unix-compressed files are allowed, except when the output file is set to NONE and the input file is overwritten
- hdfile [file name]
  Name of input Housekeeping Header Packets FITS file.
- outfile [file name]
  Name of output FITS event file. The value 'NONE' will cause the input file to be overwritten.
- (gradefile=CALDB) [file name]
  Name of the grade file. If the parameters is set to CALDB, the file is read from the calibration database.
- (addcol=no) [boolean]
  If set to 'yes', adds 'PHAS' and 'PixsAbove' columns.
- (delnull=no) [boolean]
  If set to 'yes', deletes NULL events.
- (flagneigh = no) [boolean]
  If set to 'yes', the STATUS column will be updated.
- event [integer]
  Event Threshold Level(negative value will use on-board LLD value).
- split [integer]
  Split Threshold Level(negative value will use on-board LLD value).
- (clobber=no) [boolean]
  If set to 'yes', overwrites the output file.
- (history=yes) [boolean]
  If set to 'yes', writes parameter values and other information in HISTORY blocks.
- (chatter = 2) [integer]
  Chatter Level (min=0, max=5)

B.1.4 xrtfilter

```
xrtfilter hdfile outdir [parameter=<value>]
```

xrtfilter creates a filter file, containing all the housekeeping information, to be used in the data screening. This is achived within *xrtfilter* by running in sequence *prefilter* and *makefilter*.

The first step is to derive attitude and orbit related quantities through the *prefilter* task. The satellite attitude information included in the attitude file is interpolated and the NORAD two line elements (TLEs) are propagated to determine satellite ephemeris quantities. This information is used to calculate quantities such as Bright Earth Angle, Sun Angle, Cut off rigidity, etc. The output file of *prefilter* contains the following default columns :

```
TIME,  POSITION,  VELOCITY, QUATERNION,
POINTING,  POLAR,  BORESIGHT,  SAT_ALT, SAT_LAT, SAT_LON, Z_RA,  Z_DEC,  Z_ROLL,
ELV, BR_EARTH, FOV_FLAG, SUNSHINE, SUN_ANGLE, MOON_ANGLE,  RAM_ANGLE,  ANG_DIST,
COR_SAX, SAA, SAA_TIME.
```

The default output columns are stored in a CALDB file. A different set of columns can be requested by the user via an input file supplied to the program in the parameter *outcols*. This input file is an ASCII file where column names are listed one per row. The TIME column is mandatory, and the other columns allowed are:

```
POSITION, VELOCITY, QUATERNION,
POINTING,  POLAR,  BORESIGHT,  SAT_ALT, SAT_LAT, SAT_LON, Z_RA,Z_DEC,
Z_ROLL,  ELV, BR_EARTH, FOV_FLAG, SUNSHINE, SUN_ANGLE, MOON_ANGLE,  RAM_ANGLE,
 ANG_DIST, COR_ASCA, COR_SAX, MCILWAIN_L, SAA, SAA_TIME.
```

The second step creates a filter file by running the task *makefilter*. This uses as input the output from *prefilter*, the housekeeping file and a configuration file. The *makefilter* configuration file can be an ASCII file or a FITS file and must contain for each parameter requested the following information: the parameter name, the name of the FITS file, the name of the extension containing the parameter, the interpolation method, the calibration method, the output parameter name and comments for the corresponding keyword in the output filter file. The interpolation method is used when the value of a specific parameter is not present at given times; the calibration method is applied when some simple numerical manipulations on the input HK parameters is needed. Currently, the default of the interpolation method is set to copy the last known value of that parameter. The calibration method has not been implemented yet. The default *makefilter* configuration file is part of CALDB and includes the following columns:

```
ELV, BR_EARTH, COR_SAX, SAA, SAA_TIME, ANG_DIST, FOV_FLAG,
DEC, MCILWAIN_L, MOON_ANGLE, RA, RAM_ANGLE, ROLL, SUNSHINE, SUN_ANGLE, CCDTemp, PixGtULD,
Vod1, Vod2, Vrd1, Vrd2, Vsub, Vbackjun, Baselin1, Baselin2. In addition the task adds the columns
TEN_ARCMIN, SETTLED, ACS_SAA, SAFEHOLD.
```

These columns will be included in the output filter file.

Parameters

- hdfile [file name]
  Name of the input Housekeeping Header Packets FITS file (swXXXXXXXXXXXxhd.hk).
- (nonulls=yes) [boolean]
  If set to 'yes' remove from makefilter file rows with TIME set to NULL.
- outdir [file name]
  Name of the directory for outputs.

- (clobber=no) [boolean]
  If clobber=yes overwrite the output file.
- (chatter = 3) [integer]
  Chatter Level (min=0, max=5).
- (history=yes) [boolean]
  If set to 'yes', write history keywords to the output file.
- attfile [file name]
  Name of the input attitude FITS file.
- alignfile [file name]
  Name of the input attitude alignment FITS file. Type NONE for none.
- (outfile=DEFAULT) [file name]
  Name of prefilter outfile. DEFAULT uses the standard naming convention.
- (outcols=CALDB) [file name]
  —[space-separated-values]
  Name of prefilter configuration file including the list of parameters, related to attitude and orbit information, to be included in the 'prefilter' output file. The user can also input a list of space-separated parameters between quotes on the command line.
- (orbmode=TLE_TEXT2) [string]
  Specifies the orbit mode which controls how the file, input through parameter 'orbname', will be processed. See helpfile of the 'prefilter' task.
- (orbfile=$HEADAS//refdata/SWIFT_TLE_ARCHIVE.txt) [file name]
  Name of the input orbit file.
- (leapfile=$HEADAS/refdata/leapsec.fits) [file name]
  Name of FITS leap second file.
- (rigfile=$HEADAS/refdata/rigidity.data) [file name]
  Name of the input atFunctions rigidity file.
- (origin=NASA/GSFC) [string]
  Value for FITS ORIGIN keyword.
- (interval=1) [real]
  Output interval (seconds).
- ranom [real]
  Nominal right ascension (degrees).
- decnom [real]
  Nominal declination (degrees).
- (mkfconfigfile = CALDB) [string]
  Name of the input FITS makefilter configuration file. If set to CALDB, the file is taken from the calibration database.
- (configfile = NONE) [string]
  Name of the ASCII makefilter configuration file. Type NONE to use 'mkfconfigfile' parameter.
- (hkstem=NONE) [string]
  Stem of the input Housekeeping file (to be used only if 'configfile' is different from NONE).
- (mkffile=DEFAULT) [file name]
  Name of the output makefilter file. Type DEFAULT to use the standard name.
- (tprec=0.001) [real]
  Time precision in seconds. If the difference between two times falls within the interval of [-tprec, tprec], they will be considered, by xrtfilter to be the same.

B.1.5 xrtflagpix

```
xrtflagpix infile outfile  [parameter = <value>]
```

xrtflagpix flags events occurring in bad pixels and bad column locations and events associated with the corner calibration sources. Only event files from the Swift XRT modes with positional information, Photon Counting (PC) and Windowed Timing (WT) are processed. Note that, for the WT mode only bad columns are defined and therefore flagged. For the PC mode data by using the parameter 'phas1thr', it is possible to set a threshold for the central pixel (PHAS[1]) of the 3x3 neighborhood and all pixels below that threshold are flagged.

'xrtflapix' allows for three different input files that identify bad pixels. These are:

- the Bad Pixels calibration file which includes the most up to date information about known bad pixels,
- the on-board Bad Pixels Table that includes the list of bad pixels used on-board for that observation and
- a user supplied list of bad pixels (this file has to be of the same format as the CALDB bad pixels file).

In the PC mode, events are flagged bad if any of the 3x3 neighborhood pixels are flagged bad by the on-board computer. For the PC mode the user must input also the Housekeeping Header Packet file (parameter 'hdfile') to consider the bad pixels flagged on-board which are not present in the on-ground Bad Pixels CALDB files. In the WT mode a pixel is flagged also if at least one of its nearby columns is bad.

The CCD has four calibration sources located at the corners of the detector. Events coming from these positions are flagged to allow for their identification during the screening. The locations of the calibration sources are read from a CALDB file specified in the parameter 'srcfile'.

The user has an option to choose which bad pixel list to use. All events are checked and flagged, using the information obtained from the input bad pixels files. The flag is stored as a 16 bit binary number in the column STATUS. The flag, recorded in the column STATUS, indicates if the event is considered good or bad. If the event is bad the flag, indicates why (for example dead or hot pixel) and also the origin of the information (e.g. if the bad pixel was stored in the CALDB file or the on-board Bad Pixels Table). The keyword XRTFLAG is added in the output file by the task. The list of flags is the following:

```
Events table : STATUS flags

b0000000000000000    Good event
b0000000000000001    Event falls in bad pixel from CALDB
b0000000000000010    Event falls in bad pixel from on-board Bad Pixels Table
b0000000000000100    Event falls in dead bad pixel
b0000000000001000    Event falls in hot bad pixel
b0000000000010000    Event falls in user bad pixel
b0000000000100000    Point
b0000000001000000    Column
b0000000010000000    Event has PHAS[1] < Event Threshold
b0000000100000000    Event has a neighbor bad from bad pixels list
b0000001000000000    Bad event
b0000010000000000    Event from calibration source 1
b0000100000000000    Event from calibration source 2
b0001000000000000    Event from calibration source 3
b0010000000000000    Event from calibration source 4
b0100000000000000    Saturated pixel
b1000000000000000    Flickering pixels found in the observation
```

Each event will be associated with a value that is a combination of more than one of the STATUS flags listed above. For example, if an event falls in a single pixel marked as bad in the calibration bad pixels file and also in the on-board Bad Pixels Table, the event flag will be: b0000000000100011.

If the parameter 'overstatus' is set to 'yes' (default) the STATUS column is overwritten. To update the STATUS column, without erasing the values of the previous 'xrtflagpix' run, the user must set the 'overstatus' parameter to 'no'.

All bad pixels are stored in an extension of the output file. If this already exists and the parameter *overstatus* is set to 'no', the extension is updated with new bad pixels. If requested by the user the bad pixel table is written in a separate output file. The bad pixels extension or output file contains the following columns: RAWX and RAWY give the raw coordinates of the pixel; Amp the amplifier number; TYPE which identifies whether it is a single pixel (1), a column (2) or a row (3). For columns and rows of bad pixels, RAWX and RAWY indicate the start pixel and YEXTENT the length of the set of consecutive bad pixels included in the bad pixels file. BADFLAG stores a 16 bit binary number which indicates the origin of the bad pixel with the following meaning:

```
Bad pixels table: BADFLAG flags

b0000000000000001   Bad pixels from CALDB
b0000000000000010   Bad pixels from Bad Pixels Table
b0000000000000100   Pixels dead used on-board in current observation
b0000000000001000   Pixels hot used on-board in current observation
b0000000000010000   Bad pixels in the file provided by the user
b0000000000100000   Saturated pixels
b0000000001000000   Flickering pixels found in the observation
b0000000010000000   Bad pixels found around 3x3
```

Parameters

- infile [file name]
  Name of the input event FITS file. Unix-compressed files are allowed, except when the outfile is set to NONE and the input file is overwritten.
- outfile [file name]
  Name of the output event FITS file. The value 'NONE' will cause the input file to be overwritten.
- hdfile [file name]
  Name of the input Housekeeping Header Packets FITS file (only for PC mode).
- (srcfile=CALDB) [file name]
  Name of the file containing the location of the calibration sources. This file is included in CALDB.
- (bpfile=CALDB) [file name]
  Name of the input bad pixels file. The bad pixels file is read from Calibration Database if 'CALDB' is input. Type NONE if you don't want to consider this input.
- (userbpfile=NONE) [file name]
  Name of the bad pixel file provided by the user. The default is 'NONE',i.e., no user-provided file is considered.
- (bptable=CALDB) [file name]
  Name of the on-board bad pixels file. If CALDB is input, the on-board bad pixels file stored in the CALDB is read. If NONE is input, this parameter is ignored.
- (phas1thr = 80)
  PHAS[1] threshold (min=0, max=4095) for the central pixel of a 3x3 neighborhood (only for Photon Counting Mode). Events below the threshold will be flagged.
- (outbpfile=DEFAULT) [file name]
  Name of the output bad pixels file. If outbpfile=DEFAULT, the standard naming convention is used for the output file. If outbpfile=NONE, no output file is created.

- (overstatus=yes) [boolean]
  If overstatus=yes the STATUS column and the BADPIX extension are overwritten. If overstatus=no, the STATUS column and the BADPIX extension are updated with the new bad pixels.
- (chatter = 2) [integer]
  Chatter level (min=0, max=5).
- (clobber = no) [boolean]
  If clobber=yes, the output event file and bad pixels output file will be overwritten if files with the same names exist.
- (history = yes) [boolean]
  Write parameter values in HISTORY block.

B.1.6 xrthkproc

```
xrthkproc hdfile outfile  [parameter = < value >]
```

xrthkproc processes the Swift XRT Housekeeping Header Packets file and corrects the TIME and ENDTIME columns for the frames corresponding to Windowed Timing (WT) and Photodiode (PD) modes. The time values for the TIME and ENDTIME columns are calculated with the same algorithm used to time tag the events in the WT and PD modes (see *xrttimetag* help). The algorithm requires the source position that can either provided in detector coordinates via the parameters *srcdetx* and *srcdety* or given the source RA and Dec specified through the parameters *srcra* and *srcdec*) with the RA and Dec of the nominal pointing (parameters *ranom* and *decnom*). During the slews data are taken in Photodiode (LR) and the time tag of the events assumes that all the events are at position (300,300) in detector coordinates and the input 'srcra' and 'srcdec' are ignored.

xrthkproc also calculates the values of the column HKTIME that contains the times of when the HK are measured on-board. The keyword XRTTIMES set to TRUE (T) is added to the header to indicate that the file has been already processed by the task and an extra row is added to the the 'hdfile' where the value for the column TIME set equal to the ENDTIME of the previous row.

xrthkproc can be re-run, for example for a different input source position, and the columns can be overwritten.

Parameters

- hdfile [file name]
  Name of the input Housekeeping Header Packets FITS file.
- outfile [file name]
  Name of the output FITS file. Type 'DEFAULT' to use the standard name.
- attfile [file name]
  Name of the input Attitude FITS file.
- srcdetx [integer]
  Source position given in detector coordinate DETX. If set to a negative values, the position of the source is assumed to be provided as RA and Dec and read from the parameters 'srcra', 'srcdec' and used together with the values in the parameters 'ranom', 'decnom' and 'attfile'.
- srcdety [integer]
  Source position given in detector coordinate DETY. If set to a negative values, the position of the source is assumed to be provided as RA and Dec and read from the parameters 'srcra', 'srcdec' and used together with the values in the parameters 'ranom', 'decnom' and 'attfile'.
- srcra [real]
  Source RA position (degrees).

- srcdec [real]
  Source DEC position (degrees).
- ranom [real]
  Nominal Right Ascension (degrees)
- decnom [real]
  Nominal Declination (degrees).
- (aberration=no)
  If set to no, the aberration correction is not applied to the data.
- (attinterpol=no)[boolen]
  If set to no, the attitude parameters correspond to the closest time value to the time of each frame.
- (teldef = CALDB) [file name]
  Name of the input TELDEF file. Type CALDB to use the file in the Calibration Database.
- (clobber=no) [boolean]
  If 'clobber'=yes and 'outfile'=filename, the file with the same name will be overwritten if it exists.
- (history=yes) [boolean]
  If set to 'yes', write history keywords to the output file.
- (chatter = 2) [integer]
  Chatter level (min=0, max=5)

B.1.7 xrthotpix

```
xrthotpix infile outfile  [parameter = < value >]
```

xrthotpix searches for anomalous (hot and flickering) pixels by applying statistical tests to the Swift XRT Photon Counting (PC) mode event file. The events are binned into an image and the counts in each pixel are compared to the mean counts in the whole CCD. For each pixel, the probability for its counts to be a Poisson fluctuation of the background is computed using the incomplete Gamma function Gamma(c,b), where (c) are the counts in the pixel and (b) is the background threshold evaluated using the mean of the total counts on the CCD. The parameter *impfact* allows the user to adjust this threshold. If the pixel probability is lower than a Poisson probability threshold (set through the parameter *logpos*), the pixel is considered a hot pixel candidate. The hot pixel candidates are then compared to the surrounding pixels contained in a square cell. By setting the cell size (parameter *cellsize*) of order PSF core, it is possible to discriminate a hot pixel from a pixel of the X-ray source. If the surrounding pixels have zero counts, the zero background threshold given by *bthresh* is applied. The search may be iterated by setting the parameter *iterate* to yes, but this option should be used with caution to prevent the source core from being cut out. Flickering pixels are searched for and flagged by setting the parameter *cleanflick* to yes. The algorithm is based on that implemented in the ASCA "cleansis" task. By setting the parameter 'usegoodevt' to 'yes' (default), the task searches for hot and flickering pixels on an image derived from central pixel events flagged above the threshold set by 'xrtflagpix'. All pixels classified as anomalous are flagged in the STATUS column of the event file as a 16-bit binary number (see the *xrtflagpix*). By default, the column STATUS is updated, however it is possible to overwrite the column by setting the parameter *overstatus* to 'yes. The anomalous pixels are stored as an extension in the output file. If this already exists and the parameter *overstatus* is set to 'no', the extension is updated with the new bad pixels. If requested by the user, the bad pixel table is written to a separate output file.

Parameters

- infile [file name]
  The name of the input event FITS file. Unix-compressed file are allowed, except when the output is set to NONE and the input file is overwritten.

- outfile [file name]
  Name of output event file. The value 'NONE' will cause the input file to be overwritten.
- (outbpfile = DEFAULT) [file name]
  Name of the output bad pixel file. If outbpfile=DEFAULT the standard naming convention is used for the output file. If outbpfile=NONE no output file is created.
- (overstatus = no) [boolean]
  If overstatus=yes the STATUS column is overwritten. If overstatus=no the STATUS column is updated with the new bad pixels.
- (usegoodevt = yes)
  Use only events with PHA of the central pixel (PHAS[1]) greater than the threshold set by the 'xrtflagpix' task.
- (cellsize = 5) [integer]
  Search cell size in units of pixels. Must be an odd integer greater then one.
- (impfac = 1000.0) [real]
  Value used to compute the background level (input for the incomplete Gamma function).
- (logpos = -5.3) [real]
  Logarithm of the Poisson probability threshold for rejecting a pixel. Must be negative.
- (bthresh = 3.) [real]
  Background threshold used if the candidate hot pixel's neighborhood has zero counts.
- (cleanflick = no) [boolean]
  If set to yes, search and flag flickering pixels.
- (iterate = no) [boolean]
  If set to yes, Iterate the search.
- (chatter = 5) [integer]
  Chatter Level (min=0, max=5).
- (clobber = no) [boolean]
  If set to yes, overwrite existing output file (if outfile!=NONE).
- (history = yes) [boolean]
  If set to yes, write HISTORY keywords in the output file.

B.1.8 xrtimage

USAGE

```
xrtimage infile outfile [parameter=<value>]
```

xrtimage subtracts the bias (parameter *subbias* = yes), cleans the bad and saturated pixels (parameter *cleanbp* = yes), and eliminates the calibration sources (*cleansrc* = yes) for data taken in the Swift XRT Image mode. In this mode, pixel values are proportional to the charge collected in the pixel, not to the number of events. The Swift XRT Image data are stored using the FITS Image extension. If multiple exposures (frames) are taken within an observation, these are included in a single file where each extension correspond to a single frame. 'xrtimage' processes all the extensions. It is possible to exclude frames by screening ('gtiscreen'=yes) the file for an input GTI file specified in the parameter *gtifile*.
In the Swift XRT image mode, the bias is not subtracted on board. The bias is subtracted on ground using a single value valid for all pixels of the CCD. The bias value has been evaluated from ground calibration data and it is stored in a CALDB file. This file is periodically updated by the result analysis of flight calibration data. The bias value can be input in two ways: as a single value specified through the parameter *bias* or, if the parameter *bias* is set to a negative value, the task will read the bias value information stored in the file specified through the parameter *biasfile*. If the *biasfile* parameter is set to CALDB (default) the latest version of the bias file stored in Calibration Database will be used. As an alternative, the user can input

their own bias file (with the same format of the CALDB file). The bias file can contain more than one row, each corresponding to bias measurements at different epochs. The value applied to the data is the result of bias values measured at epochs closest to the time of observation.
To clean bad and saturated pixels (if parameter *cleanbp* = yes), *xrtimage* allows three different input bad pixels list file that identify bad pixels similarly to *xrtflagpix*. These are:

- the Bad pixels calibration file which includes the most up to date information about known bad pixels,
- the on-board Bad pixels that includes the list of bad pixels used on-board for that observation and
- a user-supplied list of bad pixels (this file has to be of the same format as the bad pixels file in CALDB)

The CCD has four calibration sources located at the corners of the detector. Pixels overlapping with calibration sources location are flagged. The location of the calibration sources are read from a CALDB file specified in the parameter *srcfile*.
The output file always maintains the structure of the input file and three keywords are added or updated in the file headers. The first,'XRTPHACO', is set to 'T' to indicate that the bias correction was done, the second, 'BIAS_VAL',contains the bias value which has been subtracted, and the last,'XRTBPCNT', stores the total number of bad and saturated pixels removed.

xrtimage can be run more than once on an input file. If the input file has been already processed (XRTPHACO is set to 'T'), the task adds to all pixels the bias value previously subtracted (stored in the keyword 'BIAS_VAL') before applying the new bias correction. The bad and saturated pixels are set to NULL in the image array. The bad and saturated pixels cleaned by *xrtimage* can be output in a file by specifying the filename in the parameter *outbpfile*. This file contains as many extensions as the input file and each extension lists the raw coordinates of each pixel removed from the relative image and a flag recording why that pixel has been removed. The flag is a 16 bit binary number and the values are defined in the *xrtflagpix* helpfile.

Parameters

- infile [file name]
  Name of the input image FITS file.
- outfile [file name]
  Name of the output Image FITS file.
- (srcfile = CALDB) [file name]
  Name of the file containing the locations of the calibration sources. If set to 'CALDB' (default value), use the file in the Calibration Database.
- (biasfile = CALDB) [file name]
  Name of bias FITS file. If set to 'CALDB' (default value), use the file in the Calibration Database file.
- (bias = -40) [integer]
  Bias value. If the value is negative value, the bias is calculated using the file provided in the 'biasfile' parameter.
- (bpfile = CALDB) [file name]
  Name of the input bad pixels file. The bad pixels file is read from Calibration Database if 'CALDB' is input. If NONE is input, this parameter is ignored.
- (userbpfile = NONE) [file name]
  Name of the bad pixels file provided by the user. The default is 'NONE',i.e., no user-provided file is considered
- (outbpfile = DEFAULT) [file name]
  Name of the output bad pixels file. If outbpfile=DEFAULT, the standard naming convention is used for the output file. If outbpfile=NONE no output file is created.

- (bptable = CALDB) [file name]
  Name of the on board used bad pixels file. If CALDB is input, the on board bad pixels file stored in the CALDB is read. If NONE is input, this parameter is ignored.
- (cleanbp = yes) [boolean]
  If set to yes, remove bad and saturated pixels.
- (cleansrc = yes) [boolean]
  If set to yes, clean images from calibration sources.
- (subbias = yes) [boolean]
  If set to yes, subtract the bias.
- (clobber=no) [boolean]

  If 'clobber'=yes, the file with the same name will be overwritten if it exists.
- (gtiscreen = yes) [boolean]
  If set to yes, screen for the GTIs.
- gtifile [file name]
  Name of the input GTI file. If more than one GTI file is necessary, their list must be included in an ASCII file and input by preceeding the file name with a '@'.
- (history=yes) [boolean]
  If set to yes, write parameter values and other information in HISTORY blocks.
- (chatter = 2) [integer]
  Chatter Level (min=0, max=5).

B.1.9 xrtmkarf

```
xrtmkarf  outfile [parameter = < value >]
```

xrtmkarf generates an OGIP-style Ancillary Response Function (ARF) file which is suitable for input into the spectral fitting program XSPEC. The ARF file contains the effective area of the telescope as a function of energy needed to perform spectral analysis. This is calculated taking into account the following: mirror effective area, filter transmission, vignetting correction and optionally the PSF correction.
xrtmkarf reads the energy grid from the input RMF file (parameter *rmffile*) and calculates the arf in different ways depending on the 'inarffile' parameter. If set to CALDB an on-axius ARF is read and adjusted for the PSF and Vignetting. If 'inarffile' is set to NONE the ARF is calculated using the mirror effective area (parameter input *mirfile*) and the filter transmission (parameter input *transfile*) and a linear interpolation to adapt them to the RMF energy grid. This is then corrected for the vignetting function (always) and the PSF (optional) on the basis of the source position. The PSF correction is calculated for a point-like source taking into account the different geometry for the different operational modes of the telescope. The input files for the RMF, filter transmission, on-axis effective area, PSF and vignetting are, by default, read from the Calibration Database.

The source position in detector coordinates must be input via the parameters *srcdetx* and *srcdety*. From the input spectrum (parameter *phafile*), the extraction region is read from the WMAP extension. Only for the Photon Counting mode, *xrtmkarf* assumes the center of the extraction region to be the source position, if the *srcdetx* and *srcdety* are set to negative values.

Parameters

- outfile [file name]
  Name of the output ARF FITS file.

- (rmffile = CALDB) [file name]
  Name of the input RMF file. If set to 'CALDB' (default value), use the file in the Calibration Database, and the RMF is selected based on the information read from the spectral file.
- (mirfile = CALDB) [file name]
  Name of the input Mirror on-axis effective area file.If set to 'CALDB' (default value), use the file in the Calibration Database.
- (transmfile = CALDB) [file name]
  Name of the input Filter Transmission file.If set to 'CALDB' (default value), use the file in the Calibration Database.
- (inarffile = CALDB) [file name]
  Name of the input on-axis Ancillary Response Function (ARF) file. If set to 'CALDB' (default value), use the file in the Calibration Database. If set to NONE the on-axis effective area is calculated using the parameters 'mirfile' and 'transmfile'.
- (psffile = CALDB) [file name]
  Name of the input PSF FITS file. If set to 'CALDB' (default value), use the file in the Calibration Database.
- psfflag [boolean]
  If set to 'yes', correct for the PSF for point-like sources.
- (vigfile = CALDB) [file name]
  Name of the input vignetting FITS file.If set to 'CALDB' (default value), use the file in the Calibration Database.
- phafile [file name]
  Name of the input PHA FITS file.
- srcdetx [real]
  Source DETX coordinate[1-600]. For PC mode: if ¡0 the task assumes that the extraction region center is coincident with the source position and uses it for the ARF generation.
- srcdety [real]
  Source DETY coordinate[1-600]. For PC mode: if ¡0 the task assumes that the extraction region center is coincident with the source position and uses it for the ARF generation.
- (clobber=no) [boolean]
  If set to 'yes', the task overwrites existing output file.
- (history=yes) [boolean]
  If set to 'yes', write HISTORY keywords in the output file.
- (chatter = 2) [integer]
  Verbosity Level from 0 to 5

B.1.10 xrtpcgrade

```
xrtpcgrade infile outfile [parameter=value ... ]
```

'xrtpcgrade' calculates a single PHA and assigns an event grade for the Swift XRT Photon Counting (PC) mode data. In PC mode each event has nine PHA values associated with and stored in the column PHAS, which correspond to an array of 3x3 pixels, the central pixel and eight surrounding pixels. *xrtpcgrade* reads the nine elements of the PHAS column, calculates a single PHA value for each event and writes the result in the PHA column of the output file. The PHA value is calculated by summing all the pixels with values above split the threshold. The number of pixels above the split threshold is stored in the column named PixsAbove. The grade values are written in the GRADE column of the output file. All other columns of the EVENTS and the GTI extensions of the input file are copied unchanged to the output file.

The grades for the PC mode are assigned following the XMM-Newton pattern definition and these pattern definitions are stored in a CALDB file. The task also allows the user to assign grades following the ASCA convention, if the option *ascagrade* is set to 'yes'. In this case the grade definition is done using the routine originally developed for the ASCA task 'faint'. The routine ignores the corner pixels in the computation of the PHA value, except in the case of grade 6 where a corner pixel is included. The results of this computation are included in the two columns ASCAPHA and ASCAGRADE. This option is included only for comparison purposes and, if selected, the XRT and ASCA grades are both included in the output file. Grade selection for the Swift XRT PC data should be done using the XRT pattern convention.
xrtpcgrade add the Data Subspace keywords in the header of the event file to record the event grade (DSVALn, DSTYPn, DSFORMn). These keywords are updated by other tasks to record the screening on grades.

Parameters

- infile [file name]
  Name of the input event FITS file. Unix-compressed files are allowed, except when the output file is set to NONE and the input file is overwritten.
- outfile [file name]
  Name of output event FITS file.The value 'NONE' will cause the input file to be overwritten.
- (gradefile = CALDB)[file name]
  Name of input GRADE file. If set to CALDB (default), use the file in the Calibration Database.
- (split = -1) [integer]
  The split threshold level. If set to a negative value, use the on-board split threshold read from the input file in the keyword 'SplitThr'.
- (ascagrade = no) [boolean]
  If ascragrade=yes, the grades are calculated according to the ASCA patterns. The results are written in the 'ASCAPHA' and 'ASCAGRADE'columns.
- (history=yes) [boolean]
  If set to yes, write parameter values and other information in HISTORY blocks.
- (clobber=no) [boolean]
  If 'clobber'=yes and outfile=filename, the file with the same name will be overwritten, if it exists.
- (chatter = 2) [integer]
  Chatter Level (min=0, max=5).

B.1.11 xrtpdcorr

```
xrtpdcorr [parameter = < value >]
```

xrtpdcorr checks to see if the bias subtraction has been applied on-board to the Swift XRT Photodiode mode data and, if not, calculates and subtracts the bias. The on-board software can be set to either subtract the bias on-board before sending down the data (default) or to send down the data without bias subtraction.

The bias can be subtructed on-ground in several different ways:

- User input bias value.
  The user can input a bias value through the parameter *bias*, setting the parameter *method* to 'CONST'. If the parameter 'bias' is set to a negative value, the task will read the bias value from a file specified through the parameter *biasfile*. If the *biasfile* parameter is set to CALDB (default), the bias value is read from the file stored in the Calibration Database, if it is set to a value different from CALDB, the task will look for a file provided by the user. This file must have the same format as the bias file stored in CALDB.

- CALDB input bias value.
  (The parameter *bias* set to a negative value, *biasfile* set to 'CALDB', and the *method* set to 'CONST'). A constant bias value has been evaluated by on-ground analysis of calibration data and is stored in a CALDB file that will be periodically updated using in-flight calibration.
- Calculate bias value using a statistical method.
  A subset of telemetered PHA values is used to calculate the bias. For the Low Rate Photodiode data, the bias is calculated using the last 20 pixels below threshold telemetered in each frame and stored in the HK Header file. For the Piledup Photodiode data, the bias is calculated using the PHA values of all the pixels below threshold. *xrtpdcorr* creates an histogram of the PHA values and evaluates the bias by computing the mean value, or fitting them to a Single Gaussian or to a Double Gaussian, depending on the value of the parameter *method*. It is possible to perform a sigma clipping by setting the parameter *nclip* to a number greater than 0 and the number of sigma to clip can be input through the parameter *nsigma*. If it is not possible to perform the fit, the task uses the bias value from CALDB, giving a warning to the user that it has been done so.

After the correction the keyword XRTPHACO, in the EVENTS extension, and the BIAS extension are added or updated. The keyword is set to 'T' to indicate that the correction was done and the BIAS table stores the list of the bias values subtracted for each frame. The input event file can be supplied in the compressed format except in the case of outfile=NONE when the input file will be overwritten.

Parameters

- infile [file name]
  Name of the input event FITS.
- hdfile [file name]
  Name of the input Housekeeping header Packets FITS file. This file contains the 20 pixels used to calculate the bias for the LR. (Necessary only for the LR mode, set to 'NONE' for PU mode).
- outfile [file name]
  The output file name. The value 'NONE' will cause the input file to be overwritten.
- (bias) [integer]
  Bias value. If a negative value is input, the bias value is read from the file specified in the 'biasfile' parameter.
- (nframe = 20) [integer]
  Number of consecutive frames included in the bias calculation If set to 0 or to a negative value, all frames available are taken into account in the bias calculation.
- (nevents = 20) [integer]
  Minimum number of Photodiode events with DN value under the Low Level Discriminator included in the bias calculation when the chosen method is either MN or a Gaussian fit. (see 'method' parameter).
- (nclip = 1) [integer]
  Number of iterations to compute the Photodiode bias value. The value for this paramter is is used for the MN,SG or DG methods (see 'method' parameter). If set to 0 no clipping will be done.
- (nsigma = 2) [integer]
  Number of sigmas used in the sigma clipping algorithm (Used only if 'nclip' ¿ 0).
- (biasth = -99) [integer]
  The event threshold used to select the events to compute Photodiode bias value with MN,SG or DG methods (see 'method' parameter). If a negative value is input, the task uses the on-board Lower Level Discriminator value stored in the LLVLTHR keyword.
- (biasfile = CALDB) [file name]
  Filename containing the bias value for the Photodiode mode. If the parameter is set to CALDB (default), the bias value is read from the file stored in the Calibration Database, if it is set to a value different from CALDB, the task will look for a file provided by the user. This file must have the same format as the bias file stored in CALDB.

- method [string]
  Methods to calculate Photodiode Bias value. The methods allowed are: MN,SG,DG or CONST. 'MN' calculates a mean from the histogram generated taking number of events,input through 'nevent' parameter, under the threshold. 'SG' or 'DG' fit the histogram with a single gaussian or a double gaussian. 'CONST' uses a constant bias value taken either from the PD Bias Calibration file or from the value input in the parameter 'bias'. For the 'MN' ,'SG', 'DG' methods a sigma clipping can be applied setting the 'nclip' parameter value ¿ 0.
- (fittol = 1.E-8) [real]
  Relative tolerance of fit error computing PD bias value with SG or DG method (see 'method' parameter).
- (clobber=no) [boolean]
  If set to yes, overwrite the output file.
- (history=yes) [boolean]
  If set to yes, write parameter values and other information in HISTORY blocks.
- (chatter = 2) [integer]
  Chatter Level (min=0, max=5)

B.1.12 xrtproducts

```
xrtproducts infile outdir [parameter = <value>]
```

xrtproducts extracts high level product data files, i.e. spectra, light curves and images, for the XRT Photon Counting (PC), Windowed Timing (WT) and Photodiode (PD) modes data using as input the calibrated and screened event file (Level 2). For the XRT Image mode data, 'xrtproducts' produces a sky coordinates image plot.

- The Photon Counting mode products are: spectrum, light curve and image. The source spectrum and light curve are extracted by filtering the data on a spatial region in sky coordinates. If the parameter 'regionfile' is set to DEFAULT the user, through the parameters 'ra', 'dec' and 'radius', can specify the Right Ascension and Declination of the center and the radius of a circular extraction region. A region file with an arbitrary shape can be input through the parameter 'regionfile'. This file should have a format compatible with 'xselect' and 'ds9'. A full field of view image in sky coordinates is produced if the parameter 'imagefile' is different from 'NONE'.
- Windowed Timing mode products are: spectrum, light curve and image. The WT mode has only information in one one spatial dimension. If the parameter 'regionfile' is set to DEFAULT spectrum and light curve of the target are produced by filtering the data on a spatial region with a box shape specified by the parameters 'width', 'height' and 'roll' and centered on the source position ('ra'and 'dec' parameters). A region file with an arbitrary shape can be input through the parameter 'regionfile'. An image in RAW coordinates is produced if the parameter 'imagefile' is different from 'NONE'.
- Low rate and Piledup Photodiode modes products are: spectrum and light curve. The PD modes, Low Rate and Piledup, do not have spatial information and all the events are used to create a spectrum and a light curve.
- Image mode products are: image plot. The task 'xrtproducts' generates a plot of the full field of view image in sky coordinates.

Spectra and light curves for all modes are not background subtracted. For the PC and WT modes, the region definition is assumed to be always in sky coordinates. The user can set the binsize (in seconds) and the minimum and maximum PI channels of the extracted light curve via the parameters 'binsize', 'pilow' and 'pihigh', respectively. Moreover, the user can apply a time filter to the event file providing in input a fits GTI file (parameter 'gtifile'). If the parameter 'phafile' is different from 'NONE' the spectrum is extracted and the ARF file is calculated via the task 'xrtmkarf'. Only for PC mode, if parameter 'regionfile' is different from 'DEFAULT' the source position in detector coordinates, needed to compute the ARF file, must be given in input through the parameters 'srcdetx' and 'srcdety'. The ARF file can also be generated by running

separately 'xrtmkarf' (see 'xrtmkarf' help).
By setting the parameter 'plotdevice' to GIF or PS, the user can choose the device for the plots of the image, light curve and spectrum generated.

Parameters

- infile [file name]
  Name of the input FITS event file to be processed. Unix-compressed file are allowed
- regionfile [file name]
  Name of the region file for spatial filtering. If set to DEFAULT , a standard region is used. For PC mode the standard spatial region is a CIRCLE specified through the parameters 'ra', 'dec' and 'radius'. For WT mode the standard spatial region is a BOX specified through the parameters 'ra', 'dec', 'height', 'width' and 'roll'.
- outdir [directory name]
  Directory for the output files.
- stemout [string]
  Stem for the output files. Input 'DEFAULT' to use Standard Naming Convention.
- ra [real]
  Right Ascension (J2000.0) of the center of the region to be used for spatial filtering.
- dec [real]
  Declination (J2000.0) of the center of the region to be used for spatial filtering.
- srcdetx [real]
  Source DETX position (used for ARF file generation) (min=1, max=600).
- srcdety [real]
  Source DETY position (used for ARF file generation) (min=1, max=600).
- roll [real]
  Spacecraft roll angle in degrees. It is used to rotate the BOX shape region for spatial filtering for WT mode.
- radius [real]
  Radius of the circular region for spatial filtering in pixel (Only for PC).
- width [real]
  Width of the box shape region for spatial filtering in pixel (Only for WT).
- height [real]
  Height of the box shape region for spatial filtering in pixel (Only for WT).
- binsize [real]
  Bin size (seconds) for the light curve. If ¡0 a default value is used (10 s for PC, 1 s for WT and PD).
- (display = no) If 'yes' is input, plots the image of the field on the screen.
- lcfile [file name]
  Name of the output light curve file or DEFAULT to use stem or NONE for none.
- phafile [file name]
  Name of the output spectrum file or DEFAULT to use stem or NONE for none.
- imagefile [file name]
  Name of the output image file or DEFAULT to use stem or NONE for none.
- (gtifile = NONE) [file name]
  Name of the input GTI file for spectrum and light curve extraction or NONE for none.
- (arffile = DEFAULT) [file name]
  Name of the output ARF file. DEFAULT to use stem.

- (rmffile = CALDB) [file name]
  Name of the input RMF file. If CALDB is input (default), use the file in the Calibration Database.
- (mirfile = CALDB) [file name]
  Name of the input mirror on-axis effective area file. If CALDB is input (default), use the file in the Calibration Database.
- (transmfile = CALDB) [file name]
  Name the input filter transmission file. If CALDB is input (default), use the file in the Calibration Database.
- (psffile = CALDB) [file name]
  Name the input psf FITS file. If CALDB is input (default), use the file in the Calibration Database.
- (vigfile = CALDB) [file name]
  Name input vignetting FITS file. If CALDB is input (default), use the file in the Calibration Database.
- (psfflag = yes) [boolean]
  If set to 'yes', correct for the PSF when calculates the ARF.
- (plotdevice = gif) Device for plots (gif or ps).
- (pilow = 20) [integer]
  Minimum PI value for light curve extraction
- (pihigh = 1000) [integer]
  Maximum PI value for light curve extraction
- (clobber = no) If set to 'yes', overwrites the output files with same name if they exist.
- (chatter = 3) [integer]
  Verbosity Level from 0 to 5.
- (history = yes) [boolean]
  If set to 'yes', include history records.
- (cleanup = yes) [boolean]
  If set to 'yes', delete temporary files.

B.1.13 xrtscreen

```
xrtscreen gtiexpr exprgrade expr infile outdir createattgti createinstrgti
          gtiscreen evtscreen mkffile outfile [parameter=<value>]
```

xrtscreen allows to : i) generate a GTI file based on attitude and/or instrument HK parameters; ii) screen the data using these GTIs; and iii) screen events using a GRADE filter and/or a selection on the STATUS column. *xrtscreen* supports all the XRT science modes and requires in input the Level 1 (or 1a) event or image file and the filter file (output of *xrtfilter*).

The GTIs are calculated considering two different set of parameters, one related to the satellite attitude and ephemeris (createattgti=yes) and the other related to instrument housekeepings (createinstrgti=yes). The GTI file is created via 'maketime' and contains the time intervals where events are considered good for science data analysis. Setting the input parameter 'gtiexpr' to 'DEFAULT' and 'hkrangefile' to 'CALDB', the attitude and instrument HK parameters screening expressions are built using the standard criteria contained in the HKRANGE Calibration file. If parameter 'obsmodescreen' is set to 'yes' (default) the task adds to the GTI generation expression the selection based on the observation mode (POINTING, SETTLING or SLEW) using the columns 'SETTLED' and 'TEN_ARCMIN' of the input filter file. The user may supply attitude and/or instrument non standard criteria through the parameter 'gtiexpr' providing a boolean expression (e.g.

```
gtiexpr="CCDTemp>=-102&&CCDTemp<=-58"
```

). The calculated GTIs are used to screen the events by setting the parameter 'gtiscreen' to 'yes'. A GTI file provided by the user is also accepted in input through the 'usrgtifile' parameter.

xrtscreen allows to screen the events previously flagged bad in the STATUS column (i.e. elimination of bad pixels and calibration sources) and/or to apply a grade selection by setting the 'evtscreen' parameter to 'yes'. The standard screening criteria for GRADE and STATUS are defined in the EVTRANGE Calibration file and to use these the user should set the parameter 'evtrangefile' to CALDB and 'exprgrade' and 'expr' parameters to 'DEFAULT'.

Non standard screening criteria can be specified using the parameters 'exprgrade' and 'expr'. The first is for the selection on the GRADE column and the values can be input as a range or a single number (e.g.exprgrade=0-5 to select GRADE range between 0 and 5; exprgrade=0 to select only GRADE equal to 0). These inputs are used with the 'filter grade' command in 'xelect'. The second is for the selection on the STATUS column and the value should be input as a boolean expression, e.g. expr="STATUS==b0" to select only good events (see 'xrtflagpix' help for the definition of the values in the STATUS column).
The parameters 'gtiexpr' and 'expr' accept the expression directly from the command line or written into a text file and input by preceding the filename with '@' (e.g. expr=@file.txt). The expression in the file can be arbitrarily complex and can extend over multiple lines of the file. Lines that begin with 2 slash characters ('') are ignored and can be used to add comments.

If all the screening parameters are set to 'yes', the output events file contains only good events and the GTI extension is updated.

Parameters

- gtiexpr [string]
  Expression to generate attitude and/or instrument HK GTIs. If set to 'DEFAULT', the boolean expression is constructed using the information in the file specified with the parameter 'hkrangefile'. A text file containing the expression can be specified by preceding the filename with '@', such as '@file.txt' If the paramater is set to 'NONE', the GTI are not calculated.
- exprgrade [string]
  Expression to select the column 'GRADE' in the input event file input as a single value or a range. If set to 'DEFAULT', the string is constructed using the information in the file specified with the parameter 'evtrangefile'. If the paramater is set to 'NONE', the GRADE selection is not set .
- expr [string]
  Expression to select events using the STATUS column in the input event file. If set to 'DEFAULT', the expression is contructed using the information in the file specified with the parameter 'evtrangefile'. A text file containing the expression can be specified by preceding the filename with '@', such as '@file.txt' If the parameter is set to 'NONE', the selection on the STATUS column is not set.
- infile [file name]
  Name of the input FITS event file. Unix-compressed file are allowed.
- outdir [string]
  Name of the output directory for products.
- (clobber=no) [boolean]
  If set to 'yes', the task overwrites existing output files.
- (chatter = 3) [integer]
  Verbosity Level from 0 to 5.
- (history=yes) [boolean]
  If set to 'yes', the task writes history keywords in the output files.
- createattgti [boolean]
  If set to 'yes', the GTI file includes good time intervals based on attitude parameters.

- createinstrgti [boolean]
  If set to 'yes', the GTI file includes good time intervals based on instrument HK parameters.
- gtiscreen [boolean]
  If set to 'yes', the events file is screened for attitude and/or instrument HK GTIs generated using the parameter 'gtiexpr'.
- evtscreen [boolean]
  If set to 'yes', the events file is screened for the expressions specified in the 'expr' and 'exprgrade' parameters.
- obsmodescreen [boolean]
  If set to 'yes' the task adds to the GTI generation expression the selection based on the observation mode (POINTING, SETTLING or SLEW).
- mkffile [file name]
  Name of the input filter file.
- (gtifile = DEFAULT) [file name]
  Name of the output GTI file. If set to 'DEFAULT', the standard naming convention is assumed for the filename.
- (usrgtifile = NONE) [file name]
  Name of the user input GTI file. A text file containing a list of GTI files can be specified by preceding the filename with '@'. If set to 'NONE' (default), this parameter is ignored.
- (hkrangefile = CALDB) [file name]
  HKRANGE Calibration File Name. If set to 'CALDB' (default), the attitude and instrument HK allowed ranges are from a file in the Calibration Database. This parameter is used only if 'gtiexpr' is set to 'DEFAULT'.
- (timecol = TIME) [string]
  Name of the TIME column in the input event file.
- outfile [file name]
  Name of the output screened event file. If set to 'DEFAULT', the standard naming convention is assumed for the filename.
- (gtiext = GTI) [string]
  Name of the GTI extension in the event file.
- (evtrangefile = CALDB) [string]
  Name of the input EVTRANGE Calibration File Name. If set to 'CALDB' (default), the event selection expression is contructed from the values stored in a CALDB file. This parameter is used only if 'exprgrade' and/or 'expr' are set to 'DEFAULT'.
- (cleanup =yes) [boolean]
  If set to 'yes', the task deletes temporary files.

B.1.14 xrttam

```
xrttam  hdfile outdir [parameter = <value>]
```

xrttam calculates the corrections to the attitude file using parameters derived from the Telescope Alignment Monitor (TAM) device by running in sequence 'det2att' and 'attcombine'. The corrected attitute is applied to the XRT event data to transform the detector coordinates of the XRT instrument in sky position. After this correction, the reconstructed XRT sky positions have an accuracy of a few arcseconds.

The TAM device on-board Swift monitors the alignment between the XRT focal plane camera and the optical axis of the mirror system. The TAM system consists of a redundant pair of LEDs mounted near the XRT focal plane, an optical assembly, a mirror on the Star tracker subsystem and an optical camera which

records two images one directly reflected by the optical assembly, the other reflected by a mirror on the Star tracker subsystem.
The first image, the Primary TAM image, records the movement of the focal plane. The second image, the Star tracker image, is sensitive to both the Star tracker boresight and the XRT boresight, so that it is necessary to subtract the offset of the Primary image from the offset of the Star tracker image to isolate the contribution from the Star tracker boresight. Changes in the TAM centroid positions are related to distortions of the position of a photon on the CCD. To obtain the correct position of an event on the XRT and then in sky coordinates, the information obtained by the TAM device must be taken into account. The coordinates of the two image centroids on the TAM camera (included in the housekeeping data) are compared to the LED reference positions on the TAM (stored in the CALDB) in order to detect either distortions of the XRT structure or offsets induced by the Star tracker movements. The LEDs reference positions can be provided also by the user through the parameters 'tamrefx1', 'tamrefy1','tamrefx2', 'tamrefy2'.

xrttam translates the offsets of the TAM images into corrections to the detector coordinates on the XRT. These corrections are stored in a FITS file ('outtamfile') containing three columns with TIME, delta DETX and delta DETY information. The corrections to the detector coordinates are then transformed in corrections to the attitude, running the task 'det2att'. The 'det2att' output and the original attitude file are combined using the task 'attcombine' producing a corrected attitude file. The latter is applied to the event file when calculating sky position of the events (X and Y coordinates). The keyword 'XRTTAM' set to 'T' is added to the output corrected attitude file to prevent from applying the TAM correction more than once.

Parameters

- hdfile [file name] Name of the input FITS Housekeeping Header Packets Fits File to be processed.
- outdir [directory name] Directory for the output files.
- (tamfile = CALDB) Name of the input TAM calibration file. If set to 'CALDB'(default), use a file from the Calibration Database.
- outattfile [file name] Name of the output TAM corrected attitude file. If set to 'DEFAULT', the standard naming convention is assumed for the filename.
- attfile [file name] Name of input attitude FITS file.
- (teldef = CALDB) Name of input teldef file. If set to 'CALDB'(default), use a file from the Calibration Database.
- outtamfile [filename] Name of the output FITS file containing the corrections to the XRT detector coordinates deduced by TAM images. If set to 'DEFAULT', the standard naming convention is assumed for the filename.
- (tamrefx1 = -99) X coordinate of the reference position of the Primary TAM image. If set to a negative number, use the values stored in the file specified via the 'tamfile' parameter.
- (tamrefy1 = -99) Y coordinate of the reference position of the Primary TAM image. If set to a negative number, use the values stored in the file specified via the 'tamfile' parameter.
- (tamrefx2 = -99) X coordinate of the reference position of the Secondary TAM image. If set to a negative number, use the values stored in the file specified via the 'tamfile' parameter.
- (tamrefy2 = -99) Y coordinate of the reference position of the Secondary TAM image. If a negative value is input the value stored in the file specified by 'tamfile' parameter is used.
- (clobber = no) If set to 'yes', overwrites the output files with the same name if they exist.
- (chatter = 3) [integer] Verbosity Level from 0 to 5.
- (history=yes) [boolean] If set to 'yes', write HISTORY keywords in the output file.

B.1.15 xrttdrss

```
xrttam  [parameter = <value>]
```

xrttdrss processes the XRT image and the spectra generated on-board soon after a GRB is detected. These data are sent via TDRSS as messages and distributed first via the GCN. Because of the real time nature of the data, the messages can be processed individually or simultaneously by properly setting the input file name parameters.

IMAGE: The XRT uses the Image mode when first looks at a new GRB and generates a postage stamp image. The array transmitted contains a subset of the field of view (51x51 pixel) centered on the GRB position. The array is in RAW coordinates and the bias is not subtracted with each pixel containing the total charge detected. *xrttdrss* derives sky coordinates and subtracts the bias. It transforms the array from RAW coordinates to detector coordinates and adds the WCS keywords in the header to project the image in the sky. Note that the image is rotated respect to the celestial north. The bias is subtracted using a constant value. The bias value was derived during pre-launch calibration activities and recorded in a calibration file stored in CALDB.

In addition *xrttdrss* calculates the total flux from the source and the error circle on the source position and writes both values in keywords of the image header. The source flux is obtained by summing all pixel values in the array (a region of 4 arcmin). The flux in ergs/cm2/s is derived using a standard conversion factor, corresponding to a CRAB spectrum. Total flux and conversion factor are written in the keywords FLUX and CONVFACT respectively. The error circle on the position is derived by adding in quadrature the contributions of fours different factors: error due to the centroid calculation which depends on the source intensity, error due to the instrument alignment, error due to the attitude reconstruction and a systematic error. The first was derived from ground calibration, the other are set to pre-launch nominal values and all recorded in a calibration file included in CALDB. The resulting error circle is written in arcsec in the image header keyword ERRCTRD.

SPECTRA: After the Image mode, the XRT operates first in the Low Rate Photodiode (LR) mode and after switches to the Windowed Timing (WT) mode. Two separate spectra are calculated and sent down via TDRSS. The first is in LR, the second is either in WT (default) or a cumulative spectrum where LR and WT data are summed together. The LR data are taken on-board either with the bias already subtracted (default) or the bias not subtracted (obsolete). *xrttdrss* checks the bias and applies a constant value if not subtracted on board. The bias subtracted in the spectrum on ground was derived during pre-launch calibration activities and recorded in a calibration file included in CALDB. For the second spectrum, *xrttdrss* checks first if the spectrum contains only WT data or the cumulative LR and WT data. If the second spectrum contains only WT data no further process is necessary. Instead if the spectrum is the sum of the WT and LR, *xrttdrss* subtracts the first original LR spectrum to the cumulative (WT and LR) spectrum and recalculates the proper exposure for the subtracted spectrum. Both spectra are in PHA (not in PI) with 1024 channel (rebinned by a factor of 4 compared to original PHA array) and they are not background subtracted. Note: *xrttdrss* can operate on the second spectrum only after the first was processed. *xrttdrss* searches for a temporary file made during the processing of the first spectrum (an actual copy of that spectrum) and if not found it gives an error.

Parameters

- imagefile [file name]
  Name of the input TDRSS image file. Type 'NONE' to not process the image file.
- spec1file [file name]
  Name of the input TDRSS first spectrum file. Type 'NONE' to not process this spectrum file.
- spec2file [file name]
  Name of the input TDRSS second spectrum file. Type 'NONE' to not process this spectrum file.

- (posfile = CALDB) [file name]
  Name of the calibration file containing the different components that contribute to the error on the position. If set to CALDB, the file is read from the calibration database.
- (imbias = -1) [integer]
  Bias value for the Image mode data. If set to a negative number, xrttdrss uses the values in the file input via the parameter 'imbiasfile'
- (pdbias = -1) [integer]
  Bias value for the Photodiode mode data. If set to a negative number, xrttdrss uses the values in the file input via the parameter 'pdbiasfile'
- (imbiasfile) [file name]
  Name of the calibration file containing the bias value for the Image mode. If set to CALDB, the file is read from the calibration database.
- (pdbiasfile = CALDB) [file name]
  Name of the calibration file containing the bias value for the Photodiode mode. If set to CALDB, the file is read from the calibration database.
- (outimagefile) [file name]
  Name of the output TDRSS image file. Default set to xrt_proc_image.fits
- (outspec1file) [file name]
  Name of the output for the first TDRSS spectrum file. Default set to xrt_proc_spec1.fits
- (outspec2file) [file name]
  Name of the output for the second TDRSS spectrum file. Default set to xrt_proc_spec2.fits
- (tmpspec1file) [file name]
  Name of the spectral temporary file (created when processing the first spectrum). Default set to xrt_raw_spec1temp.fits.
- (clobber=no) [boolean]
  If 'clobber'=yes and outfile=filename, the file with the same name will be overwritten if it exists.
- (history=yes) [boolean]
  If set to yes, write parameter values and other information in HISTORY keywords.
- (chatter = 3) [integer]
  Chatter Level (min=0, max=5)
- (convfact) [real]
  Conversion factor to flux in ergs/cm2/s. The default is 2.42e-12.
- (cleanup=yes) [boolean]
  Clean all the temporary files.

B.1.16 xrttimetag

```
xrttimetag [parameter = < value >]
```

xrttimetag calculates the photon arrival time for data taken with the Swift XRT Windowed Timing (WT) and Photodiode (PD) modes and writes the DETX/DETY columns. The time tag of the events requires the knowledge of the source location on the CCD. This is usually determined using data collected when the instrument operates in imaging mode. Since the WT and PD have either a limited or no spatial resolution, it is not possible to discriminate between source and background events, and the time tagging is done assuming that all the events are from the source. The source location can be input either in detector or sky position (J2000.0) using either the 'srcdetx' and 'srcdety' or the parameters 'srcra' and 'srcdec'. The task expect by default the sky coordinates unless the parameters 'usesrcdet' is set to yes, when the detector ccordinates are expected. During the slews data are taken in Photodiode (LR) and the time tag of the events assumes that all the events are at position (300,300) in detector coordinates and the input 'srcra' and 'srcdec' or 'srcdetx' and 'srcdety' are ignored.

xrttimetag uses the readout time values stored in the ROTIME column and writes the calculated time value in the TIME column of the output file. A keyword XRTTIMES set to TRUE (T) is added to the header to indicate that the file has been already processed by the task. The columns TSTART and TSTOP in the GTI extension are updated by the task as are the keywords ONTIME, LIVETIME, DEADC and EXPOSURE.

xrttimetag calculates also for the WT and PD mode the DETX and DETY values. For the Windowed Timing mode, the RAWX coordinates are transformed to DETX using the teldef file stored in CALDB. The calculated DETX and the source DETY are stored in the DETX/DETY columns of the output file. *xrttimetag* also converts the detector coordinates into sky coordinates and writes their values in the X/Y columns of the output files. For the Photodiode mode, the input source sky position are transformed to the detector coordinates of the source. The latter are then written in the DETX/DETY columns.
The first frame taken in PD mode contains pixels which are not completely exposed. These pixels can be eliminated by setting the parameter 'npixels'. In addition, it is possible to exclude PD frames considered piled-up (parameter 'percent'), by calculating the percentage of events per frame that have a DN value above the on-board upper level discriminator. Non completly exposed pixels and piled-up frames are excluded from the GTI.

xrttimetag can be re-run on the same event file if a better source position is known and the 'TIME', 'DETX', 'DETY', 'X' and 'Y' (WT only) columns are then recalculated and overwritten. All the other columns of the EVENTS extension of the input file are copied to the output file without changes.

Parameters

- infile [file name] Name of the input events FITS file. Unix-compressed file are allowed, except when the output file is set to NONE and the input file is overwritten.
- outfile [file name] Name of output FITS file.If set to NONE, the input file is overwritten.
- hdfile [file name] Name of the input Housekeeping Header Packets FITS file.
- attfile [file name] Name of the input attitude file.
- (usehkkey=yes) [boolean] If set to 'yes', the parameters 'srcra', 'srcdec', 'ranom', 'decnom' are set to the values of the keywords 'XRA_OBJ' and 'XDEC_OBJ','XRA_PNT', 'XDEC_PNT' contained in the Housekeeping Header Packets file.
- (usesrcdet=no) [boolean]
  If set to 'no' the detector coordinates are calculated using the parameters 'srcra','srcdec', 'ranom','decnom' and 'attfile'. If set to 'yes' the detector coordinates are set to the fixed values specified through the parameters 'srcdetx' and 'srcdety'.
- srcdetx [integer]
  Source detector coordinate x. Used if usesrcdet=yes and usehkkey=no.
- srcdety [integer]
  Source detector coordinate y. Used if usesrcdet=yes and usehkkey=no.
- srcra [real] RA coordinate (J2000.0) of the source (degrees).
- srcdec [real] Dec coordinate (J2000.0) of the source (degrees).
- ranom [real] RA coordinate (J2000.0) of nominal pointing (degrees).
- decnom [real] Dec coordinate (J2000.0) of nominal pointing (degrees).
- (aberration=no)[boolen]
  If set to no, the aberration correction is not applied to the data.
- (attinterpol=no)[boolen]
  If set to no, the attitude parameters correspond to the closest time value to the time of each event.

- (teldef = CALDB) [filename] Name of teldef file. If set to CALDB, use the file in the Calibration Database.
- (npixels = 1850) [integer] Number of pixels not completely exposed to be erased from the first frame taken in the Photodiode mode.
- (percent = 50.0) [real] Percentage of events with DN value over the on-board Upper Level Discriminator within one Photodiode frame. The percentage gives an indication of pile-up in the frame. Frames considered piled-up are then not included in the good time intervals.
- (clobber=no) [boolean] If 'clobber'=yes and 'outfile'=filename the output file will be overwritten if it exists.
- (history=yes) [boolean] If set to 'yes', write history keywords to the output file.
- (chatter = 2) [integer] Verbosity Level from 0 to 5

Appendix C

ERROR CONDITION and WARNING MESSAGES

C.1 Introduction

This appendix lists the most common errors and warnings reported by the XRT tasks together with suggestions on how to recover from the errors and/or explanation for the warnings. The errors/warnings are listed by task, with those common to multiple tasks reported in the *Common* section. Errors and warnings generated by XRT tasks written in Perl are not included here.

C.1.1 Common

The following are common errors not specific to any of the tasks.

- `Error: Unable to rename temporary file.`

  In order to not corrupt the input/output file, the XRT tasks create a temporary file. This error occurs when there is not enough disk space in the directory, or the permissions are not set correctly to write in that directory.

- `Error: Unable to query CALDB.`

  The XRT tasks use CALDB to provide access to the calibration files. This error occurs because CALDB can not be accessed. If CALDB has been installed locally, check that the installation is correct and the CALDB environment variable is properly set. If CALDB is accessed remotely, check that the proper configuration file is correct and that the CALDB environment variable is set properly.

  Information on the CALDB installation and remote access are available from :

  http://heasarc.gsfc.nasa.gov/docs/heasarc/caldb/caldb_install.html

  and

  http://heasarc.gsfc.nasa.gov/docs/heasarc/caldb/caldb_remote_access.html

  respectively.

C.1.2 xrtcalcpi

- Error: XRTTIMES keyword not found or unset
  in <filename> file.
  Photon arrival times are not been calculated
  in <filename> file.
  To calculated them, please run 'xrttimetag' task on

```
'<filename>' file.
Unable to calculate PI values.
```

For the XRT Photodiode and Windowed timing modes, *xrtcalpi* should be run after the time tagging of the events. *xrtcalpi* checks that the keyword XRTTIMES, inserted by *xrttimetag*, is present in the file (and set to 'T') and, if not, reports an error.

- ```
  Error: Observation start time is not included in the validity time
         range of the GAIN file.
  ```

  The GAIN file is one of the calibration files stored in CALDB, and this error indicates that the current GAIN file does not cover the time of the observation. Users should check that the GAIN file in CALDB is up-to-date. The latest Swift XRT calibration information are available from:

  http://heasarc.gsfc.nasa.gov/docs/heasarc/caldb/swift/

  .

- ```
  Error: XRTPHA keyword not found or unset
         in '<filename>' file
         The <filename> file
         has the PHA column empty, the PI values cannot be calculated.
         To fill it, please run 'xrtpcgrade' task
         on '<filename>' event file.
  ```

  For the XRT Photon Counting mode, *xrtcalpi* should be run after the events have been graded. *xrtcalpi* checks that the keyword XRTPHA, inserted by *xrtcalpi*, is present in the file (and set to 'T'), and, if not, reports an error.

- ```
  Error: BIASONBD or XRTPHACO keyword not found or unset
         in '<filename>' file.
         No bias subtraction has been applied on PHA column,
         the PI values can not be calculated.
         Please run 'xrtpdcorr' task
         on '<filename>' event file.
  ```

  For the XRT Photodiode mode, *xrtcalpi* checks that the bias has been applied to the data via two keywords BIASONBD and XRTPHACO. The first, BIASONBD, flags if the bias has been applied on board. If not, the second, XRTPHACO, flags that the bias has been applied on ground via *xrtpdcorr*. If both are unset, *xrtpdcorr* should be run before *xrtcalpi*.

C.1.3 xrtevtrec

- ```
  Warning: the input parameter 'addcol' is set to 'no'
           but the <column name> column exists
           in <filename> file.
           The <column name> will be deleted.
  ```

  This warning is specific to two of the columns present in the event file: PHAS, containing the array to grade the events, and PixsAbove, containing the number of pixels above threshold. Whether or not these columns are present in the input file, if the 'addcol' is set to 'no', the output file will not contain them. They are unnecessary for further analysis, and might be useful for debugging purposes.

- ```
  Warning: TLMIN keywords for GRADEID column not found
           in <grade file name> file
           TLMAX keywords for GRADEID column not found
           in <grade file name> file.
  ```

  This warning is related to the CALDB file containing the GRADE definition. If users see this warning, they should check that CALDB is up-to-date. The latest calibration information is available from :

  http://heasarc.gsfc.nasa.gov/docs/heasarc/caldb/swift/

  .
  ```

- ```
  Error: BIASONBD or XRTPHACO keyword not found or unset
         in '<filename>' file.
         No bias subtraction has been applied on PHA column,
         the event reconstruction cannot be applied.
         Please run 'xrtpdcorr' task.
  ```
  For the XRT Photodiode mode, *xrtevtrec* checks that the bias has been applied to the data via two keywords BIASONBD and XRTPHACO. The first, BIASONBD, flags if the bias has been applied on board. If not, the second, XRTPHACO, flags that the bias has been applied on ground via *xrtpdcorr*. If both are unset, *xrtpdcorr* should be run before *xrtevtrec*.

C.1.4 xrtflagpix

- ```
  Warning: event with rawx=<rawx> rawy=<rawy> is out of range.
  ```
  This warning is given when pixels are flagged as bad because they are out of the nominal array space.

- ```
  Warning: Unable to find  '<ext name>' extension
           in '<bad pixels filename>' file.
           Check for 'BADPIX' extension.
  ```
  This warning is related to the file listing the bad pixels. Typically the bad pixel table is read by CALDB, but users can entered their own when running *xrtflagpix*. If users see this warning, they should check that CALDB is up-to-date or that the user-supplied table conforms to the bad pixel table format as stored in CALDB. The latest calibration information and their file format are available from :

  http://heasarc.gsfc.nasa.gov/docs/heasarc/caldb/swift/

  .

- ```
  Error: Unable to read Calibration sources information
         Unable to get Calibration Sources Position.
  ```
  This error is related to the calibration file containing the position of the Calibration sources. Users should check that CALDB is up-to-date (the latest calibration information is available from http://heasarc.gsfc.nasa.gov/docs/heasarc/caldb/swift/).

- ```
  Error: Unable to read 'BaseLine' value
         from <header packet filename> file.

  Error: The <hd filename> file is not appropriate for
         <filename> events file.
  ```
  These errors occur when the HK file containing the packet frame headers (*hd*), and the science event file are not for the same time interval. Users should check that the TSTART and TSTOP keywords in the HK file include the times that are in the event file.

C.1.5 xrthkproc

- ```
  Warning: The source is out of the CCD for the frame number <frame number>.

  Warning: Unable to calculate times for this frame.

  Warning: Unable to calculate times for all the TIMING modes' frames.
  ```
  These warnings are given when the source position is outside of the CCD area and the time for a frame cannot be calculated. The last warning is given when, for all Timing modes' frames, the times could not be calculated and the HK output file has the values set to -1 for all rows corresponding to the TIMING modes. Users should check that the attitude file is correct for this observation and/or that the latest telescope definition file is available in CALDB.

- `Warning: CCD Frame Time: <frame time> is not included in the time range of the <attitude filename> Attitude file.`

  This warning is given when the HK file, containing the packet frame headers (*hd*), and the attitude file do not cover the same time interval. Users should check that the TSTART and TSTOP keywords in the attitude file include the times that are in the HK file.

- ```
  Error: <RA_NOM || DEC_NOM> and <RA_PNT || DEC_PNT> keywords not found
         in <hd filename> file.
         Please use <ranom || decnom> input parameter.

  Error: <RA_OBJ || DEC_OBJ> keywords not found
         in <hd filename> file.
         Please use <srcra || srcdec> input parameter.
  ```

  xrthkproc needs the pointing direction coordinates. These are read from the header of the input HK file (*hd*) using RA_NOM and DEC_NOM or RA_PNT and DEC_PNT or from the input parameters 'ranom' and 'decnom'. This error indicates that keywords RA_NOM and DEC_NOM or RA_PNT and DEC_PNT are not found and the user has to re-run the task by entering the pointing direction coordinates via the 'ranom' and 'decnom' parameters.

- `Error: <Nominal || Source> <RA || DEC> value is out of valid range.`

  This error indicates that the coordinates (source or pointing) that were entered were out of the valid range. The valid range for RA and DEC is 0-360 and -90-90 degrees respectively.

- ```
  Error: Unable to calculate source SKY coordinates.

  Error: Unable to calculate source DET coordinates.
  ```

  These errors occur for different reasons: either the attitude file is not appropriate for the HK input file (check if the time ranges of the two files overlap), the teldef file is incorrect (check that CALDB is up-to-date), or the input RA and DEC coordinates are incorrect.

C.1.6 xrthotpix

- `Warning: pixel rawx=<rawx value> rawy=<rawy value> is out of range. Ignored.`

  xrthotpix checks that each pixel has their RAW coordinates (which are the telemetered values) within the allowed range (0-599) and gives this warning if some pixels are out of the allowed range.

- ```
  Error: 'cellsize' parameter value is not valid.
         'cellsize' parameter value must be > 1.
         'cellsize' parameter must be odd.

  Error: 'logpos' parameter value is not valid.
         'logpos' parameter value must be < 0.
  ```

  These errors occur when the input parameter 'cellsize' or 'logpos' are entered with a value which is not expected. Users should check the input values.

- `Error: Unable to build counts image.`

  xrthotpix works on Photon Counting mode data and this error is given if the input event file has a format different from that expected. Users should check that the file is for the Photon Counting mode (header keyword DATAMODE='PHOTON').

- ```
  Error: Unable to search hot pixels.

  Error: counts image is empty.
  ```

  These errors occur when the input event file is empty. Users should check that the outputs produced by previous tasks were non-empty files.

- `Error: Unable to flag hot and flickering pixels`
  `       in <filename> temporary file.`

  The task writes an extension containing the hot and flickering pixels detected. This error occurs when there is not disk space available to add this additional extension to the event file.

C.1.7 xrtimage

- ```
  Warning: GTI range is empty.

  Warning: All images are out of the GTIs range
           All images are rejected
           no output file will be created.
  ```

  xrtimage allows screening if an image exposure is within the time interval where the instrument is working with nominal parameters (GTI). These warnings are printed only if the parameter 'gtiscreen' is set to 'yes'. The first warning is given when there are no GTIs in the file, and the second when all the exposures taken in image mode are outside of the GTI and therefore have been rejected.

- `Warning: all input  parameters flag set to 'no'. Nothing to be done!`

  The following input parameters 'cleanbp', gtiscreen', 'cleansrc' and 'subbias' set how *xrtimage* operates on the input image. If all are set to 'no', *xrtimage* can not perform any of these operations.

- `Warning: all screen parameters set to 'no' and bias value is 0. Nothing to be done!`

  This warning occurs when the input parameter 'subbias' is set to yes and 'cleanbp', gtiscreen', 'cleansrc' are set to 'no', but the calculated value for the bias is 0. The result is that *xrtimage* can not perform any calculation (similar to the previous warning).

- `Warning: Calculated bias value is 0.`

  The calculated bias value is 0 and nothing is thus subtracted from the input image. This occurs when the bias CALDB file does not have a value for the time interval of the observation. Users should re-run the task and input the bias value in the parameter 'bias'.

- ```
  Error: The times in the
         <filename> file
         are not included in the GTIs.
         <filename> file range is:
         TSTART = <tstart value> and TSTOP = <tstop value>
         GTI range is:
         TSTART = <gti tstart value> and TSTOP = <gti tstop value>
  ```

  This error occurs when the times in the input file do not match the time in the GTI. This may indicate that the GTIs are not appropriate for this file. To recreate a new GTI, users should run *xrtfilter* and *xrtscreen*.

- `Error: Unable to get Calibration sources information`
  `       but 'cleansrc' set to 'yes'.`

  This error is related to the calibration file containing the positions of the Calibration sources. Users should check that CALDB is up-to-date (the latest calibration information is available from http://heasarc.gsfc.nasa.gov/docs/heasarc/caldb/swift/).

- ```
  Error: This task does not process Image with <coordinate system> coordinate system.
         Valid coordinate system is: RAW.
  ```

  The input image file for *xrtimage* should be in RAW coordinates (see the header keywords CTYPE). If the file is in DET or sky the task gives this error.
  ```

C.1.8 xrtmkarf

- `Error: Unable to calculate vignetting correction.`

  ```
  Error: Unable to calculate PSF correction.
  Error: Unable to apply PSF correction.

  Error: Unable to  correct Mirror Effective Area.

  Error: Unable to calculate filter transmission value
  ```

  These errors are related to the calibration files containing the vignetting and PSF coefficients, the mirror effective area and the filter transmission. They indicate that either CALDB was not found or that the format of the files are not as expected. Users should check that their CALDB setting was correct and up-to-date. The latest calibration information is available from :

  $$http://heasarc.gsfc.nasa.gov/docs/heasarc/caldb/swift/$$

  .

- `Error: Source coordinates are out of range.`

  The source position is an input parameter to *xrtmkarf* and should be entered in DET coordinates. This error is given when the position is outside of the expected range (1-600).

- ```
  Error: Unable to read WMAP
       : in <pha filename> file.
  ```

  xrtmkarf uses the extraction region for the appropriates modes, written in the primary header WMAP of the input spectral file to calculate the ARF. This error occurs when the input spectral file does not contain the WMAP. Users should re-extract the spectrum via *xselect*.

C.1.9 xrtpcgrade

- ```
  Warning: TLMIN keywords for GRADEID column not found
           in <grade file name> file
           TLMAX keywords for GRADEID column not found
           in <grade file name> file.
  ```

  This warning is related to the CALDB file containing the GRADE definition. If users see this warning, they should check that CALDB is up-to-date (the latest calibration information is available from http://heasarc.gsfc.nasa.gov/docs/heasarc/caldb/swift/).

- ```
  Error: The multiplicity of the PHAS column is: <col multiplicity>
         but should be: 9
  ```

  xrtpcgrade works on Photon Counting mode data and this error is given if the input event file has a format different from that expected. Specifically the column PHAS should contains an array of 9 values. Users should check that the file is for Photon Counting mode (header keyword DATAMODE='PHOTON').

C.1.10 xrtpdcorr

- `Warning: Unable to create bias histogram.`

  This warning is given when there are not enough pixels/events to calculate the bias. For the Low Rate Photodiode, *xrtpdcorr* uses the 20 pixels stored in the HK file containing the science header packets. For the Piled-up Photodiode, it uses the events below threshold found in the event file. Users can try to change the input parameters, decrease the 'nevents' or 'biasth', increase 'nframe' or use a different method to calculate the bias.
  ```

- `Warning: There are only <number of events>  events to build bias histogram`
  `but the minimum number requested is: <min number of events>`
  `Unable to fit bias histogram with a gaussian distribution.`

  This warning is given when the number of events for the histogram is too small. Users can change the minimum number of events using the parameter 'nevents' or change the threshold via the parameter 'biasth'. Alternatively , they can change the method for how the bias is calculated via the parameter 'method'.

- `Error: The times in the`
  `<filename> file are not included in the range time`
  `of the <hd filename> file.`

  This warning is given when the HK file, containing the packet frame headers (*hd*), and the event file, does not cover the same time interval. Users should check that the TSTART and TSTOP keywords in the event file include the times that are in the HK file.

- `Error: Multiplicity of the LRBiasPx column is: <mul value>`
  `but value allowed is: 20`

  This errors indicates that the HK file is not what is expected by the task. The input HK file, containing the packet frame headers (*hd*) should contain a column, LRBiasPx, with an array of 20 elements.

C.1.11 xrttimetag

- `Error: Unable to calculate GTIs range.`

  `Warning: GTI range is empty.`

  xrttimetag assigns the time to each event and calculates the GTI. This error and warning are given when the GTI in the output file could not be calculated because some of the input parameters are not compatible with the observation: for example, the attitude is from a different observation or the teldef is not appropriate for the input coordinates.

- `Warning: source position is out of range.`

  `Warning: The source is out of the CCD for the frame <ccdframe>`

  The RA and DEC of the source position is an input parameter to *xrttimetag*. This warning is given when this position translated in detector coordinates is outside of the expected range of the DET coordinates (1-600).

- `Warning: Found <n. of pixels out of CCD> of <total n. of pixels>  pixels out of CCD`
  `<% of pixels out of CCD> pixels will be rejected.`

  This warning gives a summary of how many pixels are found outside of the CCD. These pixels have the value in the TIME, DETX and DETY columns set to -1 and for Windowed timing also the X and Y column. The location of the source in detector coordinates depends on the attitude, if the percentage is high, users should check that the values for the source and pointing position parameters were entered correctly.

- `Warning: CCD Frame Time: <time> is not included`
  `in the time range of the`
  `<attitude filename> Attitude file.`
  `Setting detx = <default detx> and dety = <default dety>`

  This warning is given when the HK file, containing the packet frame headers (*hd*), and the attitude file do not cover the same time interval. Users should check that the TSTART and TSTOP keywords in the attitude file include the times that are in the HK file.

- `Error: Unable to calculate X and Y from the source and pointing coordinates.`

  This error is given when the input coordinates for the source and pointing position may have been entered incorrectly by the users: for example, they are out of bounds.