Data Reduction Method of the Follow-up X-ray Telescope onboard Einstein Probe

EP-FXT Science Data Center [1]

[1] Key Laboratory for Particle Astrophysics, Institute of High Energy Physics, Chinese Academy of Sciences, 19B Yuquan Road, Beijing 100049, China

Abstract: We present the Follow-up X-ray Telescope Data Analysis Software (FXTDAS, version 1.05), a software package that has been designed for calibration and screening data obtained by the Follow-up X-ray Telescope (FXT) onboard the Einstein Probe (EP). Data modes and data files of FXT that are related to data reduction are also introduced. This software package is built based on the HEASOFT package (version 6.33) and fully compatible with the HEASOFT. Additionally, the CALDB (version 1.05), containing calibration data of the FXT, has been released with the FXTDAS. The data reduction procedure for FXT includes correction of time, elimination of the events caused by particles, computation of the PI values, identification of abnormal pixels and columns, and reconstruction of events. Generic tools from HEASOFT, such as Xselect and Xspec, can also be used to manipulate the data files of FXT.

Key words: Data Products, Data modes, Data analysis, pipeline

1 Introduction

The Follow-up X-ray Telescope (FXT) is one of the main payloads of the Einstein Probe (EP), a mission led by the Chinese Academy of Sciences (CAS). It consists of two X-ray mirror modules (FXTA and FXTB), each containing 54 nested Wolter-I paraboloid-hyperboloid mirror shells. And each module is equipped with a PN-CCD as the focal plane detector. The PN-CCD has an integrated image and a frame-store area, both with 384 × 384 pixels.

The primary goals of FXT are: (1) discover and characterize X-ray transients, particularly faint, distant and rare X-ray transients; (2) identify X-ray outbursts from otherwise normally dormant black holes; (3) search for X-ray sources associated with gravitational-wave events and precisely locate them; (4) conduct deep follow-up observations of targets discovered by EP. To achieve these goals, FXT is designed to be capable of localizing an X-ray source, measuring the energy spectra with high energy resolution in X-ray band 0.3–10 keV, and in some case, performing observations with high time resolution. Table 1 summarizes the FXT parameters.

Table 1. The FXT characteristics.

FXT parameters	Value
Energy range	0.3–10 keV
Mirror	Wolter-I
Detector	PN-CCD
Effective area	$600\,\mathrm{cm}^2$ @1.25 keV (on-axis)
Field of view	$60'$
Detector elements	384×384
pixel scale of integrated area	$75\mu\mathrm{m}\times75\mu\mathrm{m}$
pixel scale of frame-store area	$75\mu\mathrm{m}\times51\mu\mathrm{m}$
PSF	$<= 30''$
Sensitivity	10 μCrab, $2.4\times\mathrm{E}^{-13}$ erg cm^{-2} s^{-1}

In order to complement their capabilities, the PN-CCD of FXT offers three readout modes (science modes). The conventional X-ray readout modes are the Full Frame mode (FF) and Partial Window mode (PW). In these modes, photons are registered in the integrated image during the integration time and then transferred to the frame-store area during the readout time. The FF and PW modes are designed for weak source (< 5 mCrab for FF; < 200 mCrab for PW), but provide a better position identification. While, the Timing mode (TM) is utilized for very bright sources (> 200 mCrab) and for high time resolution.

EP satellite was successfully launched on January 9, 2024 from the Xichang Satellite Launch Centre in China. To better utilize of the FXT data, we will introduce the FXT data and its analysis software.

2 FXT modes

The FXT is responsible for quick follow-up observations of the triggered sources by EP and it can automatically select the science mode to adjust the source count rate. Additionally, it has the capability to observe interested targets, with the science mode for a given observation can be commanded from the ground segment via the command data uplink route. The three main science modes are FF, PW and TM mode.

The FF mode provides full imaging but is limited in time resolution. A full field of view is accumulated every 49.8848 ms, with a frame-transfer time of about 0.1152 ms. Each CCD frame is rapidly transferred into a frame-store area and then read out by three CAMEXs (CAMEX_L, CAMEX_M and CAMEX_R). Onboard processing includes bias subtraction and application of lower level discriminator, but event reconstruction is not performed.

In PW mode, only the pixels in the central 128×61 area (162–222 rows) controlled by the middle CAMEX (CAMEX_M) are read out. The sub-image is accumulated every 1.9319 ms and then transferred to frame-store area by 227 serial clock shifts (about 0.0681 ms). Between two adjacent frames, there is a 200-microsecond pause in order to discharge the electric charge.

The TM is designed for very bright sources and high time resolution. This mode performs consecutive row readouts and the shift clock, which corresponds to the readout of one row, is about 23.68 μs. However, after reading out 384 rows (a frame), there is a pause time (reset time) of about $6\times23.68\,\mu+200\,\mu$s. Photons from the source are uniformly distributed along the readout direction in this mode, but there is a count enhancement at the position of the source during the reset time. This mode does not record the arrival time of each photon, but it records the frame time and the order of photons at the CAMEX_M. Therefore, algorithms are required to calculate the arrival time of each photon.

The images of the FF and PW modes are both a 2D histogram of the number of events, whereas the image of the TM mode is not a 2D but only 1D along the X coordinate. In TM mode, the X in the CCD coordinate contains the spatial information, while Y is a counter incremented by one when a row is read out. Fortunately, the short reset time can lead to an enhancement of Y, which can be used to determine the position of the source (by creating a 2D image). Table 2 presents the science modes and their characteristics.

Table 2. The FXT science modes and these characteristics.

Mode	Image capability (X×Y)	Time resolution	Flux level	Integrated time	Readout time	Reset time
FF	384 × 384, 2D	50 ms	< 5 mCrab	49.8848 ms	0.1152 ms	-
PW	128×61, 2D	2.2 ms	< 200 mCrab	1.9319 ms	0.0681 ms	0.2 ms
TM	128×384, 1D	23.68 μs	> 200 mCrab	-	23.68 μs	$6\times23.68\,\mu+200\,\mu$s

3 Data files

The raw FXT data is converted into FITS files and organized in directory structures, known as the Level 1 data product. Each observation data identified by an observation ID implies a stable satellite pointing direction. For each observation, there are orbit file, attitude file, event files and a filter file (mkf file). The mkf file contains the time-histories of various parameters to which a list of selected Good Time Intervals (GTI) can be produced. For scientific users, data analysis starts from Level 1 data product. The root directory of every product is named by the Observation ID and is organized as fllowes:

- auxil directory
  - attitude file
  - orbit file
- fxt directory
  - event directory
    - event files

- hk directory
  - mkf file

For the event files, there is one file per readout mode, and the file name format for the event files uses the convention: fxt_[module]_[obs_id]_[mode]_[filter]_[pp]_[uf]_[ext]_[lev].fits.

The filename contains several keywords:

- [fxt] is a prefix to indicate the mission name.
- [module] indicates the instrument (a/b) of FXT.
- [obs_id] contains an 11 digits number to identify the observation ID.
- [mode] indicates the scientific mode to obtain the data (ff/pw/tm).
- [filter] gives the thickness of the filter of FXT, and it includes thin, thick, open and so on.
- [pp] identifies if the event data were taken with the satellite in pointing mode (po) or during a slew (sl).
- [uf] indicates the file is an unfiltered file.
- [ext] indicates the type of this file, and always set to 'evt'.
- [lev] gives the file version.

The file version definition rule is as follows: (1) represented by three characters; (2) the first character indicates data integrity (0, incomplete; 1, complete); the second character indicates the data version (the number of times the data is produced under a certain software version); the third character represents the software version; (3) the third character has the highest priority, followed by the second character; (4) the second character is represented in order of priority by a-z, then A-Z, and finally 0-9; (5) the third character is only represented in the order of priority by a-z; (6) once the data is complete, all products are based on complete data, and there will be no situation such as 1aa<0ba.

4 FXT Data Analysis Software

The data reduction of FXT uses tools that account for the calibration, screen and algorithms specific as well as generic tools, Xselect, used to manipulate the FITS data files. The FXT data can be downloaded from EP archive, and the Follow-up X-ray Telescope Data Analysis Software (FXTDAS) and the CALDB can be downloaded from web site http://epfxt.ihep.ac.cn/analysis.

The FXTDAS is designed to achieve FXT data analysis processing and extract scientific products, such as energy spectra, images, light curves, ancillary response files (ARF), redistribution matrix files (RMF) and background files. The software is written in ftools style and is fully compatible with the HEASOFT software.

The software interfaces with the FXT CALDB, which is a collection of files essential for accurate data reduction and analysis. These files provide detailed information about the properties and performance of the PN-CCD, mirror and instrumental components for each module. It is organized into a well-defined directory structure, making it easy for users to locate and access the calibration files required for their specific analysis tasks.

4.1 FXTDAS Installation

The FXTDAS package is constructed on the basis of the HEASOFT package, version 6.33. Users can install, compile, and utilize this software just like you would with HEASOFT.

The source code of FXTDAS could be compiled and installed on the Linux Distribution such as CentOS. Python version 3.8 or higher is required. Other installation requirements can be the same as those for the HEASOFT installation.

4.1.1 Source code Installation

The downloaded software package (fxtsoftv1.05.tar.gz) can be extracted using the tar and gunzip commands as follows:

```
gunzip fxtsoftv1.05.tar.gz
tar zxvf fxtsoftv1.05.tar
```

Once the package has been extracted, the next step is to configure the software. The configure script will scan the system for libraries, header files, compilers, etc., and then generate the Makefile. If the configuration is successful, the software can be built and installed in sequence. These steps are outlined as follows:

```
cd fxtsoftv1.05/BUILD_DIR
./configure --prefix=INSTALL-PATH
make
make install
```

If the "make install" step was successful, users can proceed with initializing the software. The following are example commands assuming the Bourne shell, which users should modify to be appropriate for your system:

```
export HEADAS=INSTALL-PATH/(PLATFORM)
source $HEADAS/headas-init.sh
```

However, these commands will only work for the current terminal session. Users can add them to .bashrc file for them to be available in all terminal sessions. Thus, users can then initialize the software envirnment using the fxtinit command.

```
export HEADAS=INSTALL-PATH/(PLATFORM)
alias fxtinit="source $HEADAS/headas-init.sh"
```

4.1.2 Docker Installation

From docker hub, one can pull the image using the following command:

```
docker pull generl/fxtsoft:v1.05
```

Alternatively, one can download the image from FXT website using the command:

```
docker load -i  fxtsoftimagev1.05.tar
```

And then,

```
docker run  -v D:\docker\caldb:/caldb -v D:\docker\home:/home -it generl/fxtsoft:v1.05  /bin/bash
```

4.2 CALDB Installation

The CALDB does not need to be installed and can be unzipped to any convenient location. As long as users set the correct location in the environment variables, it will work fine. Users can enter the following commands or add them to your .bashrc file.

```
export CALDB=/PATH/CALDB
export CALDBCONFIG=$CALDB/caldb.config
export CALDBALIAS=$CALDB/software/tools/alias_config.fits
```

For the docker installation, the CALDB package is unzipped to host path of CALDB (e.g., D:/docker/caldb)

4.3 Data Reduction

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

- Stage 1
  The processing of FXT data involves time correction of events (only TM mode), elimination of the events caused by particles (currently only for FF and PW modes), computation of the PI values, identification of bad/hot pixels and columns, reconstruction of events.

- Stage 2
  The GTI file is obtained by setting conditions on mkf files. The GTI file, as well as selections on the GRADE and STATUS columns are as inputs to Xselect tool.

FXTDAS provides several tasks with each task is to accomplish a step of data analysis. These tasks in Table 3 are run in sequence for different data modes.

Table 3. The tasks provided by FXTDAS

Task name	FF mode	PW mode	TM mode	Function
fxttimecorr	-	-	✓	Time correction of events
fxtparticleidentify	✓	✓	-	Elimination of the events caused by particles
fxtpical	✓	✓	✓	Computation of the PI values
fxtbadpix	✓	✓	✓	Flag of bad pixels/columns
fxthotpix	✓	✓	✓	Identification of hot pixels/columns
fxtgrade	✓	✓	✓	Reconstruction of events
fxtgtigen	✓	✓	✓	Generation GTI files
Xselect	✓	✓	✓	Extract cleaned files, spectra, light curves and images
fxtexpogen	✓	✓	✓	Generation of exposure files
fxtarfgen	✓	✓	✓	Generation of ARF
fxtrmfgen	✓	✓	✓	Generation of RMF
fxtbary	✓	✓	✓	Barycentric correction

4.3.1 FF and PW modes

- fxtparticleidentify: This task is used for the observations of FF and PW modes, as well as some observations of TM mode with weak sources.

  fxtparticleidentify evtfile=/epfs/data/Science/202403/08500000024/fxt/event/fxt_a_08500000024_ff_01_po_uf_evt_0ba.fits outfile=fxt_a_08500000024_removalParticle.fits

- fxtpical: This task calculates the Pulse Invariant (PI) values for event files obtained by FF/PW/TM modes. The PI values are corrected for average positional gain variation (Charge Transfer Inefficiency, CTI) and for changes of the gain with time.

  fxtpical evtfile=fxt_a_08500000024_removalParticle.fits outfile=fxt_a_08500000024_pi.fits

- fxtbadpix: This task flags events occurring in bad pixels of the FXT detectors. The task allows for three different input files that identify bad pixels: on-ground bad pixel calibration file which includes the most up to date information about known stable bad pixels; the Disabled Pixel File which includes the list of pixels disabled by on-board processing for the current observation; a user supplied Bad Pixel File (with the same format as the on-ground CALDB bad pixel file).The value of the 'STATUS' column specifies the quality of the flagging (i.e. if the event falls on a stable bad pixel from the on-ground CALDB File or on an on-board disabled pixel).

  fxtbadpix evtfile=fxt_a_08500000024_pi.fits outfile=fxt_a_08500000024_bad.fits

- fxthotpix: This task searches for anomalous (hot and flickering) pixels by applying statistical tests to the FXT event file. For each detector of the Focal Plane Module, and for each time interval defined by the input parameter 'binsize', the events are binned into images. For each image, hot/flickering pixel are identified by comparing the counts in each pixel to the mean background counts.

  fxthotpix evtfile=fxt_a_08500000024_bad.fits outfile=fxt_a_08500000024_hot.fits

- fxtgrade: This task calculates the PI values and assigns the grades for the FF, PW and TM modes. Each event has nine PI values associated, stored in the column PIS, corresponding to an array of 3×3 pixels, the central pixel and eight surrounding pixels. This task reads the nine elements of the PIS column, calculates a single PI value for each event and writes the result in the PI column of the output file. The PI value is calculated by summing all the pixels with values above the split threshold (indicted by the parameter pithresh).

  fxtgrade evtfile=fxt_a_08500000024_hot.fits outfile=fxt_a_08500000024_grade.fits

- fxtgtigen: This task will create a Good Time Interval (GTI) FITS file based on parameters in a Make filter (MKF) data FITS file.

  fxtgtigen hkfile=/epfs/data/Science/202403/08500000024/fxt/hk/fxt_08500000024_mkf_1ca.fits module=fxta expr=NONE outfile=fxt_a_08500000024_gti.fits

- Xselect: It is a tool to screen the data interactively and also to extract the high-level products. Within Xselect it is possible to set and apply the following types of filters:

  - filter grade 0-12
  - select event "status==b0"
  - filter time file fxt_a_08500000024_gti.fits
  - extract event
  - save event fxt_a_08500000024_clean.fits
  - extract spectrum
  - save spectrum fxt_a_08500000024_spec.fits
  - filter pha_cutoff 73 925
  - extract curve
  - save curve fxt_a_08500000024_lc.fits
  - extract image
  - save image fxt_a_08500000024_img.fits

  Additionally, to create a region filter (filter region src.reg), users have to examine the image and then select the region so as to include the source of interest. An easy way is to use ds9 to select the source region and background region.

  A tool fxte2pi can be used to get PI and the input parameter is energy (e.g., fxte2pi 0.5, a PI value at energy 0.5 keV will be returned).

- fxtexpogen: This tool generates exposure maps that accounts for CCD bad pixels and columns, attitude variations and telescope vignetting.

  fxtexpogen evtfile=fxt_a_08500000024_clean.fits
  mkffile=/epfs/data/Science/202403/08500000024/fxt/hk/fxt_08500000024_mkf_1ca.fits outfile=fxt_a_08500000024_expo.fits

- fxtarfgen: This tool generates an ARF suitable for imaging observations with the FXT. This ARF characterizes the effective area as a function of energy for a specific observation and instrument setup, quantified in units of [cm**(2) counts/photon].

  fxtarfgen specfile=fxt_a_08500000024_spec.fits expfile=fxt_a_08500000024_expo.fits outfile=fxt_a_08500000024_arf.fits

- fxtrmfgen: This tool generates a RMF file appropriate for analysis of FXT data. The RMF encapsulates the mapping between the physical properties of incoming photons (such as their energy) and their detected properties (such as detector pulse heights or PHA) for a given detector. This mapping is stored in the form of a 2-D matrix.

  fxtrmfgen specfile=fxt_a_08500000024_spec.fits outfile=fxt_a_08500000024_rmf.fits

4.3.2 TM mode

The TM mode is designed for very bright sources and for high time resolution but does not have 2D positional information, which leads to spectral distortion when only PI correction is done. Thus, it is important to do position correction, in which all photons are assumed to originate from one position. While the event time is assigned by the frame time and its arrival time has a relationship with its RAWY in PN-CCD, which is only a counter incremented by one when a row is read out. Thus, the source position should be known by users.

The fxttimecorr corrects the event time for TM mode based on the RAWY values stored in the events. It must be run before fxtpical, because it will provide the true y-coordinate for fxtpical to do gain and CTI correction. If there are multiple sources in the field of view, only the coordinates of one source can be input at a time, and the gain and CTI correction can only be applied correctly to this source. These sources can be run through this task in sequence to obtain the correct correction for each source.

fxttimecorr evtfile=/epfs/data/Science/202401/13600000087/fxt/event/fxt_a_13600000087_tm_03_po_uf_evt_0aa.fits outfile=fxt_a_13600000087_timecorr.fits attfile=/epfs/data/Science/202401/13600000087/auxil/ep_13600000087_att_1ba.fits ra=39.54154192 dec=25.86643826

The other analysis steps are similar to the FF and PW modes, and these description will not be repeated here.

4.3.3 Barycentric Correction

fxtbary applies barycenter correction to FXT event data files. The task reads EVENTS and GTI extensions in event files, and overwrites the TIME column for EVENTS extension (START and STOP columns for GTI extension). While the keywords TSTART, TSTOP, TIMESYS and TIMEREF for both extension are changed.

fxtbary evtfile=fxt_a_08500000012_clean.fits outfile=fxt_a_08500000012_bary.fits orbitfile=ep_08500000012_orb_1da.fits ephemfile=/epfs/data/software/fxtsoftv1.02/fxtsoftv1.02/fxt/tasks/fxtbary/de421.bsp ra=83.63321666666667 dec=22.01446388888889

4.3.4 Pipeline tool

An convenient way to conduct data reduction of FXT is to utilize the fxtchain tool. This tool allows users to set various parameters, many of which are already set to default values. The fxtchain script is capable of searching an observation directory for the input files required to carry out the individual tasks in the data reduction process.

The fxtchain process scans the input directory to ensure that both event data and ancillary data are present. It then sequentially performs the following tasks for each event file(limited to ff, pw, and tm modes): fxttimecorr (only applicable in TM mode), fxtpical, fxtbadpix, fxthotpix, fxtgrade, fxtgtigen, xselect, fxtexpogen, fxtarfgen, and fxtrmfgen (with parameters enabled for fxtexpogen, fxtarfgen, and fxtrmfgen tasks). The output files of these tasks are stored in the specified output directory.

- The basic fxtchain uses the following required input parameters:

  fxtchain indir=/epfs/data/Science/202401/13600000087 outdir=13600000087
  where 'indir' is the directory name where the data files are located, 'outdir' is the output directory.

  By default, fxtchain will generate the clean event files and GTI files. However, for the TM mode, the position and gain correction are always incorrect. In order to obtain an accurate correction, users must set the source coordinates as follows:

  fxtchain indir=/epfs/data/Science/202401/13600000087 outdir=13600000087 ra=39.54154192 dec=25.86643826

- The following command shows how to obtain high level data products such as energy spectra.

  fxtchain indir=/epfs/data/Science/202401/06800000423 outdir=06800000423 pattern=all

- The following command shows how to obtain FXT module a/b files.

  fxtchain indir=/epfs/data/Science/202401/06800000423 outdir=06800000423 pattern=all modula=a

- In this example, fxtchain only runs a data mode:

  fxtchain indir=/epfs/data/Science/202401/06800000423 outdir=06800000423 pattern=all datamode=FF

- This example shows how to extract a specify source region and its backgrouond region:

fxtchain indir=/epfs/data/Science/202401/06800000423 outdir=06800000423 pattern=all datamode=FF srcffregion=src.reg bkgffregion=bkg.reg module=a

The region parameters related to the mode are: srcffregion, srcpwregion, srctmregion, bkgffregion, bkgpwregion and bkgtmregion.

5 New releases

This document is primarily for fxtsoft version 1.05. After the on-orbit testing is completed, we will provide a more comprehensive software version and documentation with more accurate results.

Please note that FXT is currently undergoing in-flight calibration. The derived results may be subject to larger uncertainties, so please use them with caution.

Acknowledgements

References