DEPARTMENT OF SPACE RESEARCH
BIRMINGHAM UNIVERSITY
Asterix Data Analysis
ASTERIX Programming Note 016

PROG/016

DJ Allan
23 Jul 1994

The ASTERIX FITS Interface

Contents

1 Introduction

The Asterix FITS interface is the major new feature of a program of work making the Asterix data interface more flexible. Some of the other highlights of this work which will not be discussed further in this note, include,

- A greatly simplified programming interface – all the important routines fit on one side of A4 (in a modest sized typeface).
- A decoupling of the file interface from ADAM. Programs can be written stand-alone rather than as A-tasks if required.
- Improved compatibility between Asterix and other Starlink software using the NDF.

2 Which FITS Formats?

The FITS formats which will be supported first by the Asterix interface will be those defined by the OGIP calibration memos, ie. the formats for time series, spectra, images and events as created by XSELECT.

What does this mean in practice? Data products created by XSELECT can be read directly into Asterix programs. Asterix copes quite easily with datasets without quality or errors attached where it makes sense to do so (such as XIMAGE format images).

Moving data from Asterix NDFs to FITS is more of a problem due the lack of certain features in the OGIP definitions. Rather than invent new FITS standards ourselves the default Asterix behaviour will be simply to ignore those NDF components which cannot be represented in FITS. Thus we cater for the user who wants to use software accessing FITS files conforming to the OGIP standards, but don't encourage him to create datasets which can't be read by anything else but Asterix.

The one exception to the above rule will be the graphics control information. Asterix applications will be able to write graphics control data into an extension of a FITS file which can then be used by GDRAW or the image processing.

Other FITS formats such as the MIDAS formats could be incorporated into the interface system if the demand was sufficient.

3 Programming Issues

3.1 How does it work?

A substantial document nearing completion describes in detail how the data interface work is designed and implemented. Suffice to say that data is not *converted* to HDS, but that both FITS and HDS formats have been incorporated into the same logical data model and the operations by the application on that model map on to different ways of accessing the underlying files.

The main challenge of the FITS interface is not in getting one working, but doing so efficiently and and in a manner which does not require the way Asterix applications interface to their data to be radically altered.

3.2 Choosing the Format

Asterix applications will support a number of ways of specifying the format of output files. In those cases where the output is an obvious derivation of the input (the majority) then the file format will be inherited from the input unless the user overrides it. Additional environment settings will allow the user to, for example, choose HDS or FITS as the output format regardless of the nature of the inputs.

The syntax for specifying parts of HDS files remains as it is at present. The format for specifying FITS keywords and extensions is borrowed from that of FTOOLS. Some examples are,

```
myfile                  ; the whole FITS file - .fits is assumed
myfile.pha[1]           ; whole name of file needed, use first
                          FITS extension only
myfile.pha+1            ; Alternative version of above
myfile.pha[1]%fits      ; as above if ambiguous representation
                          (not very likely!)
myfile.pha^BITPIX       ; specific keyword
myfile.pha[3]^NEVENTS   ; specific keyword in extension
```

3.3 Data Access

One part of the programmability problems of Asterix in the past has been the requirement to use file mapping in analysis tasks. Thus puts off the casual programmer because the requirement to split even simple tasks into two layers to dereference pointers. The new data interface provides three data access mechanisms,

- The conventional `map/unmap` interface similar to that in Asterix at the moment. Still very useful and easy to program for many classes of application.
- A simple `put/get` interface. Often a user wants to write a quick bit of hack Fortran, often accessing the data into arrays of fixed size, mucking about with it and then writing it back to the file.
- A buffered data access. Meant for use by certain classes of application, especially those which access event data or binned data in a vectorised form (such as ARITHMETIC). Aims to reduce the amount of memory required by such programs working on large datasets.

The important point for the user is that all 3 access methods will be supported on all data formats where appropriate, although the most efficient method will vary depending on the way the data is accessed and the way it is stored on disk.

4 Examples

Before plunging into the mappings from FITS formats to Asterix interface items a couple of lower level mappings relevant to all FITS files are now given.

Anywhere the user can currently enter an HDS scalar primitive value, a FITS keyword name will also be accepted. Likewise anywhere an HDS array can be supplied a FITS image extension or FITS bintable column can be supplied instead.

Two examples for binned data are given, for spectra and images as well as the mapping for event data.

5 Spectra

FITS files with the HDU keywords `HDUCLASS` and `HDUCLAS1` set to `OGIP` and `SPECTRUM` respectively will be accepted by Asterix as both a general 1-D binned dataset, and in particular as a spectrum. The following mapping exists between FITS header keywords and their equivalents in the current Asterix system.

FITS keyword	Asterix Equivalent
NAXIS2	Number of bins in the spectrum
HDUCLAS2	Data exposure corrected?
POISSERR	Poissonian errors? If true then Asterix can take variance=data
OBJECT	Header target name
TELESCOP	Header observatory name
INSTRUME	Header instrument name
FILTER	Header filter name
EXPOSURE	Header exposure time component
BACKFILE	Background reference object
RESPFILE	Asterix energy response
ANCRFILE	"
AREASCAL	"
DATE-OBS	Header BASE_MJD object
TIME-OBS	Header BASE_UTC object
DATE-END	Difference from above gives header OBS_LENGTH component
TIME-END	"
RADECSYS	No equivalent but no FK4 data?
EQUINOX	Header equinox component
RA_PNT	Header component AXIS_RA
DEC_PNT	Header component AXIS_DEC
PA_PNT	Header component POSITION_ANGLE (but see below)

- The comment of the PA_PNT keyword reads "file average of ROLL". The definitions of roll angle and position angle are different in Asterix and checks need to be made to decide what is actually being used here.

- Most of the other keywords present in a FITS spectrum are for documentary purposes and have no relevance to Asterix when reading the files. Some of them would be filled in by Asterix writing those files, such as the `CREATOR` and `HISTORY` keywords.

The XSPEC fields `CHANNEL` maps to the spectrum axis, `COUNTS` to the data and `QUALITY` to Asterix quality bits. Grouping information will cause a rebinning of the data, errors and energy response on the fly to take place initially, as Asterix does not support grouping – a situation which could change in the long term.

The good time intervals extension (GTI) maps on to an equivalent structure in the Asterix instrument structure.

6 Images

FITS files with the HDU keywords `HDUCLASS` and `HDUCLAS1` set to `OGIP` and `IMAGE` respectively will be accpted by Asterix as both a general 2-D binned dataset, and in particular as an image. In addition, any FITS primary array of two dimensions will be similarly recognised. Only the differences between the image format and the spectral format are listed below.

FITS keyword	**Asterix Equivalent**
NAXIS1	Size of image X axis
NAXIS2	Size of image Y axis
CRPIXi	Spaced axis origin and scale
CDELTi	”
RA-NOM	Header component AXIS_RA
DEC-NOM	Header component AXIS_DEC
DROLLANG	Header component POSITION_ANGLE (converted)
PHA-LOW	Asterix sort box components
PHA-HIGH	

Asterix will support both simple arrays and scaled arrays.

7 Event Tables

FITS files of `CONTENT` set to `BASIC`, or extensions with the HDU keywords `HDUCLASS` and `HDUCLAS1` set to `OGIP` and `EVENT` respectively will be accepted by Asterix as event datasets. The field names will be the same as those in the FITS file and map to existing Asterix event structures. Most Asterix event processing applications would need some alteration to operate efficiently on FITS event data, but a scheme has been devised for doing this.

Instrument specific keywords in the primary header map are equivalent to identically named components in the Asterix instrument structure.

8 Summary

The first useful result of this work will be an upgrading of the Asterix application AST2XSP which converts Asterix spectra to XSPEC spectra. This will be upgraded to write the new FITS spectral file format (also handling associated background spectra), *and* will convert in the reverse direction.

Shortly after AST2XSP would follow AST2XCR and AST2XIM to convert between Asterix and XRONOS and XIMAGE formats. These three applications (or rather three faces of one application) would serve as test beds for the Asterix interface work and provide an interim solution for those requiring data reformatting. It is estimated that these programs will be available within 2 to 3 months. Asterix applications would then start to be converted with intention that most would be done by the next release late in the year.