ASTRONOMY & ASTROPHYSICS
SUPPLEMENT SERIES

MAY II 1994, PAGE 53

Astron. Astrophys. Suppl. Ser. **105,** 53-55 (1994)

The FITS image extension

J.D. Ponz[1], R.W. Thompson[2] and J.R. Muñoz[3]

[1] ESA/ECNOD Villafranca, P.O. Box 50727, 28080 Madrid, Spain
[2] CSC, Code 684.9, GSFC, Greenbelt, MD 20771, U.S.A.
[3] ESA/SSD ESTEC, P.O. Box 299, 2200 AG Noordwijk, the Netherlands

Received October 8; accepted November 8, 1993

Abstract. — The Flexible Image Transport System (FITS) image extension provides a convenient means of storing two or more related but not necessarily identical multidimensional arrays in a single file. It conforms to the guidelines and rules for generalized FITS extensions and is structured in a manner identical to that of a primary header and data unit; thereby making it simple to interpret and compatible with existing FITS readers. The structure of the image extension and its potential uses are discussed.

Key words: astronomical data bases: miscellaneous techniques: image processing

1. Introduction

It is often useful to store separate but related astronomical data sets as one file to be transported or archived as a single entity. A common example is an array of data points with a corresponding array of weights. The two data sets need to be closely connected yet may also be entirely different in terms of dimensions, dynamic range, physical units or pixel format. The Flexible Image Transport System (FITS) format has evolved since its definition (Wells et al. 1981) into the standard file format for astronomical data transfer and storage. Data sets such as those described above however, could not be easily stored in FITS format. The available choices were to merge the data sets into a non-homogeneous primary array, store them in random group format (Greisen & Harten 1981) which has since become a deprecated standard, store all the data as ASCII characters in a table extension (Harten et al. 1988), or store the data as a series of one dimensional arrays in a binary table extension. Note that although a multidimensional array convention for the binary table extension has been formulated (NOST 1991), it is not part of the current binary table proposal.

The image extension was first proposed for use by the International Ultraviolet Explorer (IUE) project, as a means for storing SEC Vidicon image data with associated data quality information (Muñoz 1989). A proposal for its adoption as a standard FITS extension was submitted in 1992, and was formally approved by the European FITS committee in May of that year. It was later approved by the Working Group for Astronomical Software of the American Astronomical Society in 1993. The resolution will be submitted to IAU Commission 5 for adoption at the IAU General Assembly meeting to be held in 1994.

The image extension is basically a replication of the primary header and data unit (HDU) following the guidelines and rules for FITS extensions (Grosbøl et al. 1988). As such, the image extension becomes the most basic extension possible in FITS format. Unlike other extensions which are frequently used as the primary data storage structure (i.e., by setting NAXIS = 0 in the primary header), the image extension is intended to provide a logical format for storing supplementary information associated with the primary data unit. When used in this manner, the resulting FITS file has a structure which is consistent with the nature of the data being stored. This format provides the following features:

- Allows the storage of an unlimited number of multidimensional arrays.
- Arrays are stored in separate extensions, thereby allowing each array to have its own header, semantic contents and dynamic range.
- FITS readers can easily skip over separate extensions.
- No additional keywords or conventions need to be adopted.
- Any new conventions adopted for the primary HDU would be compatible with the image extension as well.
- Only minor software changes should be required to allow existing FITS readers to handle image extensions.

- Extracting the image extension as a separated file produces a simple FITS primary array without any further processing, other than replacing the XTENSION = 'IMAGE 'keyword with SIMPLE = T.

2. Header format

The image extension header begins in the first byte of the first logical record following the primary data array or the previous extension. Although allowed, it is recommended that the primary header does not set the keyword NAXIS = 0, since it would not make sense to extend a non-existing image with another image. It is also required that the primary header of the FITS file have the keyword EXTEND set to T.

The extension header format is basically identical to the primary header format as described in the definition of FITS (NOST 1991). The major differences are that the SIMPLE keyword is replaced with XTENSION = 'IMAGE ', the keywords GCOUNT and PCOUNT are added, and the BLOCKED and EXTEND keywords allowed in the primary header are not allowed in the extension. The content of the header should be sufficient to allow a FITS reader to decide if it should read or skip over the extension. The card images in the header of the image extension use the following keywords, in the order indicated in Table 1:

- XTENSION The first required keyword, it is used to describe the type of extension. The value field shall contain the character string 'IMAGE '.
- BITPIX Integer value specifying the number of bits used to represent each data value. All BITPIX values allowed in the primary header for the primary data matrix are also allowed for the image extension header. The absolute value of this variable is used in computing the size of the data structure with Eq. (1).
- NAXIS Non-negative integer value no greater than 999, representing the number of axes in the data array.
- NAXISn Indexed keyword with a non-negative integer value, representing the number of positions along the n-th axis of the data array. The keyword NAXISn must be present for all values n = 1, 2, ..., NAXIS.
- PCOUNT Non-negative integer value defining the number of parameters per group in the associated random-group data structure (Greisen & Harten 1981). An image extension shall have PCOUNT = 0.
- GCOUNT Non-negative integer value defining the number of groups in the associated random-group data structure. An image extension shall have GCOUNT = 1.
- END This keyword has no associated value. Columns 9-80 shall be filled with ASCII blanks.

The guidelines and rules for the FITS extensions (Grosbøl et al. 1988) establish that all extension headers must incorporate the PCOUNT and GCOUNT keywords, in order to permit random-groups data structures to be written within the extension. The size in bits of the extension data is determined as follows:

$$S = |B| \times G \times (P + N_1 \times N_2 \times ... \times N_{\rm NAXIS}) \quad (1)$$

where S is the extension size in bits, B represents the value of the keyword BITPIX, G is the value of the keyword GCOUNT, P is the value of PCOUNT and N_n is the value of the indexed keyword NAXISn for n = 1, 2, ..., NAXIS.

Table 1. Keywords for Primary HDU and image extension

Primary HDU	Image Extension
SIMPLE	XTENSION= 'IMAGE '
BITPIX	BITPIX
NAXIS	NAXIS
NAXISn	NAXISn
EXTEND	PCOUNT = 0
...	GCOUNT = 1
(other keywords)	...
...	(other keywords)
END	...
	END

Table 2. Reserved keywords for the image extension

Extension	Documentation	Observation	Array
EXTNAME	AUTHOR	DATE-OBS	BSCALE
EXTVER	REFERENC	TELESCOP	BZERO
EXTLEVEL	COMMENT	INSTRUME	BUNIT
	HISTORY	OBSERVER	BLANK
		OBJECT	CTYPEn
		EQUINOX	CRPIXn
		EPOCH	CROTAn
			CRVALn
			CDELTn
			DATAMAX
			DATAMIN

Additional keywords, located in the extension header after the GCOUNT keyword and before END, may be used to describe history of the data, characteristics of the observations, characteristics of the data array and other auxiliary information. These keywords are optional but, when used to describe an extension, they shall have the meaning given in the definition of the FITS format (NOST 1991). Table 2 summarizes these reserved keywords. The first column shows the keywords specifying the name, version and level of the extension (Grosbøl et al. 1988), the second column lists the keywords used for data documentation, the third column lists instrumental and observation related keywords and the fourth column includes data array related keywords.

3. Data records

The data format is identical to that of a primary data array. This format will allow each image extension to contain a single data array of 1-999 dimensions with a data structure and scale factors independent of other arrays.

The image extension data shall begin in the first byte of the first record following the last record of the extension header. The first value of each subsequent row of the array shall immediately follow the last value of the previous row. In this manner, the array structure is independent of the record structure. Arrays of more than one dimension shall consist of a sequence such that the index of the first dimension varies most rapidly, and the index of the last dimension varies least rapidly.

Table 3. Example of primary and image extension headers

```
        Primary data header
0........1.........2.........3.........4.........5...
12345678901234567890123456789012345678901234567890...
SIMPLE  =                    T / Standard FITS format
BITPIX  =                   32 / 4-Byte integers
NAXIS   =                    2 / Number of axes
NAXIS1  =                  768 / Pixels per row
NAXIS2  =                  768 / Number of rows
EXTEND  =                    T / Extension present
CTYPE1  = 'SAMPLE  '             / X axis
CTYPE2  = 'LINE    '             / Y axis
BUNIT   = 'COUNTS  '             / Pixel units
...
END
        Image extension header
0........1.........2.........3.........4.........5...
12345678901234567890123456789012345678901234567890...
XTENSION= 'IMAGE   '             / IMAGE extension
BITPIX  =                    8 / 1-Byte integers
NAXIS   =                    2 / Number of axes
NAXIS1  =                  768 / Pixels per row
NAXIS2  =                  768 / Number of rows
PCOUNT  =                    0 / Parameters per group
GCOUNT  =                    1 / Number of groups
CTYPE1  = 'SAMPLE  '             / X axis
CTYPE2  = 'LINE    '             / Y axis
BUNIT   = 'UNITLESS'             / No units
...
END
```

4. Example

The image extension can be used to encode data structures in a simple way. A typical example of such a structure is a CCD image and the model of the associated PSF. The CCD image has a given size and sampling steps, while the modelled PSF could have different ones. This data structure can be efficiently stored as a single file by encoding the CCD image as the primary data array and storing its associated PSF as an image extension.

Another case of interest is to associate weights or quality factors with each pixel in an image. In this case the dimensions of the primary image and its associated quality data are the same but their physical units, dynamic ranges or pixel representations may be different. The example included in Table 3 shows the primary data header for an image together with the associated image extension header. Image pixels are encoded as 32-bit, twos complement signed integers, while the pixel quality flags stored in the image extension are encoded as 8-bit unsigned integers.

5. Conclusions

The image extension provides a simple method for adding supplementary data to the primary data array, and for storing multiple multidimensional arrays in a single FITS file. These features make the image extension a useful addition to the FITS format.

Acknowledgements. The authors would like to thank P. Grosbøl and B. Schlesinger for the detailed comments on this paper and for their involvement during the testing phase of this format.

References

Greisen E.W., Harten R.H. 1981, A&AS 44, 371
Grosbøl P., Harten R.H., Greisen E.W., Wells D.C. 1988, A&AS 73, 359
Harten R.H., Grosbøl P., Greisen E.W., Wells D.C. 1988, A&AS 73, 365
Muñoz J.R. 1989, ESA IUE Newsletter 32, 12
NOST 100-0.3b NASA Draft Standard, 1991, Definition of the Flexible Image Transport System, NASA/OSSA Office of Standards and Technology, GSFC
Wells D.C., Greisen E.W., Harten R.H. 1981, A&AS 44, 363