ASTRONOMY & ASTROPHYSICS
SUPPLEMENT SERIES

JUNE **1988,** PAGE 359

Astron. Astrophys. Suppl. Ser. **73,** 359-364 (1988)

Generalized extensions and blocking factors for FITS

P. Grosbøl (1), R. H. Harten (2), E. W. Greisen (3) and D. C. Wells (3)

(1) European Southern Observatory, Karl-Schwarzschildstr. 2, D-8046 Garching, F.R.G.
(2) The Netherlands Foundation for Radio Astronomy, Postbus 2, NL-7990 AA Dwingeloo, the Netherlands (now at RCA Astroelectronics, Princeton, NJ, U.S.A.)
(3) National Radio Astronomy Observatory, Edgemont Road, Charlottesville, VA 22903-2475, U.S.A.

Received July 17, accepted September 25, 1987

Summary. — A general design for future extensions to the FITS tape format is defined. The design preserves compatibility with existing FITS tapes and software, including the « random groups » and other existing extensions of FITS, but its generalized design will permit a wide variety of new types of extension files to be designed in the future. Rules for the blocking of FITS logical records are also defined.

Key words : data processing — tape format — data transport.

1. Introduction.

The FITS tape format standard (Wells *et al.*, 1981, hereafter « Basic FITS ») was developed to transfer regularly gridded astronomical image data between different locations. It has been implemented by most of the major observatories of the world and has been endorsed by working groups for software in both Europe and North America. The design of Basic FITS included several provisions which were intended to permit the format to be extended in order to transmit new kinds of data structures. The « random-groups » extension of FITS (Greisen and Harten, 1981) exploited these possibilities to produce a tape format design which is useful for transmitting data which is regularly gridded on some axes and irregularly gridded on other axes. It has been implemented by several radio observatories for the transfer of radio interferometer visibility measurements. This extension has been recommended by the North American working group for use by North American observatories. The FITS tape format was recommended (resolution C1) for use by all observatories by Commission 5 at the 1982 meeting of the IAU at Patras (*IAU Inf. Bull.* No. **49,** 1983). Note that the General Assembly of the IAU adopted (resolution R11) the recommendations of its commissions, including the FITS resolution.

There are two motivations for creating FITS extensions. First, the ability to develop new types of extensions enables the astronomical community to satisfy the need to be able to transmit new types of data structures, while still adhering to the same set of basic rules. The need to represent and transfer data structures which are not images or single data matrices was apparent from the time of the original FITS agreement in 1979, and has grown steadily, especially in recent years. The second motivation for the use of extensions is that they allow us to transfer collections of related data structures in an organized manner, i.e., extensions provide a simple relational data base capability. For example, the new tables extension agreement means that tables, lists, etc., associated with a data matrix can now be written in a single FITS file in a manner which implicitly establishes the relationship between the different pieces of information.

Several FITS extension formats have been designed already and more are expected to be devised. It is possible that some of them might conflict with each other or with the random-groups format which has already been endorsed as a standard. This observation led to the realization that there is a need for a general set of rules to govern the design of all future FITS extensions. During several months of discussions such a set of rule evolved and it is presented in this paper. A companion paper (Harten *et al.*, 1988) presents a tables format as an example of the application of the rules.

During recent years it has become clear that a significant number of the major data producing institutes regard the physical block length of Basic FITS (2880

Send offprint requests to : P. Grosbøl.

bytes) as inefficient both in terms of the length of tape and number of I/O operations needed for writing a given amount of data. Further, the rapid increase of the amount of digital data being obtained by modern detectors (e.g. CCD's and radio correlators) emphasizes this problem. The present generation of computers provides memory sizes in the Mbyte range. The majority of astronomical sites can therefore, without significant problems, read tapes with physical blocks larger than 2880 bytes. The usage of the FITS format on new data carrying media (e.g., local area networks (LANs), cartridge tapes, floppy and optical disks) will also require agreements on the physical block sizes to be used. It seems therefore appropriate to adopt the general concept of blocking for use with the standard FITS format.

The European FITS Committee and the Working Group for Astronomical Software of the American Astronomical Society have agreed, by formal resolutions adopted in 1986, to implement the generalized extension and blocking rules which are documented in this paper, effective on 1987 January 1. The two groups will jointly offer these resolutions to IAU Commission 5 for adoption when the Commission meets in Baltimore in 1988.

2. Records and blocks : terminology.

Two definitions must be made in order to avoid possible confusion in terminology in this paper and in future papers on the FITS data format standard. The term « record » will refer to the basic 2880-byte FITS unit or piece of information ; this corresponds to the concept of a « logical record ». A FITS file will consist of an integral number of such records, and extensions will always begin with a new record. The term « block » will refer to the physical block size on magnetic tape or other media. The reader should note that these two terms were used interchangeably in the previous FITS papers.

3. New record-blocking rules.

The logical record size of FITS format data is, and will remain, 23040 bits (2880 8-bit bytes). However, after 1987 January 1, the physical block size used for transport of data fields in FITS format on a data carrier is given as a blocking factor times this logical record length. The last block of a data file shall be truncated to the minimum number of logical records required to hold the remaining data, in accordance with the ANSI X3.27-1978 specifications. Note that the last logical record of a FITS file must still be padded with zeroes or spaces to 2880 bytes, as specified by the rules of the original FITS agreement. By default, the blocking factor is one for all media. An allowed range or set of blocking factors can be specified, by agreement, for a given data carrying medium. Files written with an allowed blocking factor will be regarded as conforming to the FITS standard. No changes to the FITS header keywords are required for implementation of this agreement. In particular, the current blocking factor should never be specified by a header keyword value.

For nine-track magnetic tapes conforming to the ANSI X3.40-1983 specification, blocking factors from one to ten (28800 bytes per block), inclusive, will be permitted in interchange. This significant change in the FITS standard is commonly referred to as « Long-Blocks FITS ». Although this change was chiefly motivated by the desire to increase the packing efficiency of 6250 bpi tapes (ANSI X3.54-1976), the agreement also applies to the 800 (ANSI X3.22-1973) and 1600 bpi (ANSI X3.39-1973) densities.

Old FITS readers will often be unable to decode blocked FITS tapes. To make it possible for users of old decoding programs to determine the reason for such errors, a new logical keyword BLOCKED is defined to indicate if a check on the blocking factor used is required. Although not required, the keyword (*always* with value true, BLOCKED=T) is recommended as an indication that the file *may* be written with a blocking factor different from one. When used, the BLOCKED keyword *must* be specified within the first logical record of a file so that it will appear in a listing of the first record. Note that if old FITS files are reblocked during copy operations, they may appear in interchange *without* the BLOCKED keyword ; this implies that all new reading programs should be prepared to deblock *any* FITS tape they read, whether or not the BLOCKED keyword is present.

4. Basic philosophy.

The most important rule for designing new extensions to FITS is that existing FITS tapes must remain valid. We are not permitted to alter the basic format in such a way as to make existing FITS tapes invalid or unreadable by standard FITS tape reading programs. This does not mean that the FITS format cannot evolve or change. To avoid this trap, FITS was deliberately designed to be capable of evolving. Two rules of Basic FITS form the basis for all existing extensions, and for the new rules defined in this paper.

— Any number of 2880-byte records may appear *after* the records which transmit the primary data matrix. These appended records have often been called « special records ». The rule obliges all Basic FITS reading programs to be prepared to skip over such records if they are not programmed to interpret them.

— FITS files may be written in which there is no primary data matrix, either because the number of axes is set to zero or because the product of the dimensions of the axes is zero.

Simply stated, all FITS extensions must appear after the main FITS header and its associated primary data array and each extension should begin at the start of a new 2880-byte record.

All existing tapes contining extensions to Basic FITS conform to these rules of Basic FITS and therefore are valid « FITS tapes », even though they contain data structures which are not simple binary matrices. The existing random-groups format is the best-known example of such an extension. The new rules defined in this paper also conform and they systematize the format of the special records so that new types of extensions can be devised freely by implementors of FITS software. The new rules allow users to create new extensions with a high degree of protection from conflicts with extensions devised by other implementors, and without obsoleting either the Basic FITS standard or the existing extensions.

The basic structure of a FITS tape is quite flexible. By adding new keywords and data axes, users can design a data structure to suit their needs. In the past, users have created entire customized header structures using the HISTORY and COMMENT keywords. These structures were valid in the FITS format, readable by other users and did not require the approval of a standards committee. The set of rules for extensions to FITS presented in this paper is designed to provide a framework in which users can continue to create new data structures to suit their local needs, while still following the FITS standard.

The only restriction that will have to be placed on the freedom to create new extensions is that there should be only one approved extension format for each type of data organization. New extension types have to be created whenever the organization of the information is such that it cannot be handled by one of the existing extension types. It will be the function of each user who creates a new extension type to check with the standards committee to see if an extension already exists for that type of data organization and to propose one if it is really a new extension type. The contents of an extension and the optional keywords used, etc. will depend on the particular application. Thus one can use a standard table extension for all type of tabular information, without having to define a new extension type. With this restriction in mind, users should feel free to create new extension types when the need arises, but should always be aware that use of non-standard extensions will inhibit the interchange of astronomical data.

5. Guidelines and rules for FITS extensions.

Before we can specify the details of the extensions to FITS, it is necessary to discuss the basic guidelines and requirements for extensions. These fall into two broad classes. The first concerns those requirements which maintain the compatability and flexibility of the existing FITS standard. The second contains those new features which are desirable to solve the problems for which the extensions are needed. A list of the guidelines and requirements is given below.

— Existing FITS tapes, including those with existing standard extensions, must be compatible with the new extension standard. FITS files which contain combinations of standard and new extensions must be allowed in order to facilitate the transition to the new design.

— The presence of a new extension in a FITS file should not affect the operation of a program which does not know about the new type of extension.

— Only the binary and character coding conventions specified in the FITS standards should be used in FITS extensions. At present, these are printable ASCII in headers, and 8-bit unsigned, and 16-, and 32-bit twos-complement integers without « byteswap » in data matrices. The FITS committees expect to add IEEE floating point notation to this set by agreements in the future.

— Extensions should have the same structure as the Basic FITS file, a header plus information. The extension data structures should be self-defining and readable by both humans and machines. The same basic rules for creating FITS headers should apply to extension headers, i.e., they will contain a required subset of standard keywords, consist of ASCII text and may have any length. This will allow one to reuse the code which interprets the primary FITS header.

— A program scanning a tape should be able to locate the beginning of any extension and should be able to skip over the extension, i.e., to find the start of the next one. This requirement implies that the extension header must specify in some consistent and standardized manner the total number of data bits which are associated with it.

— It should be possible for extensions to FITS to support hierarchical structuring of various types of data entities. One needs to be able to transmit tables, etc., which are associated with the basic data matrix and to maintain the relationship between the tables and the data. The ability to specify structures more than one level deep is included in order to provide a framework for future developments.

— It must be possible to devise new types of extensions without prior approval. This implies that *keywords* in the primary FITS header may not be used to announce the existence of a particular type of extension, because these would need to be approved by the standards committees.

— It should be possible to append any number of extensions to a primary header and it associated data matrix.

— If there is more than one type of extension in a FITS file then the extensions may appear in any order.

— Anyone wishing to create a new extension format is free to do so, but should check with the FITS standards committee to insure that there is no conflict in the extension type naming and that the proposed format conforms to the rules for FITS extensions.

— No information in either the primary header or any extension header should ever explicitly refer to the physical block sizes of FITS blocks on tape or any other medium. This principle, which was followed in the design of both Basic FITS and the random-group format, enable FITS files to be copied between media and reblocked with no changes to the information content of the FITS files.

The above list places a set of minimum requirements on features which must be built into FITS to be able to handle extensions in a systematic manner. The primary requirements are the requirement for a header at the beginning of each extension and the ability for a program to be able to identify the type of an extension or to skip over it even if it does not recognize the type.

6. An addition to the main FITS header.

Tapes which conform to this standard are required to include the keyword EXTEND in their main header immediately after the last required keyword of the Basic FITS specifications. The value field should be the logical value true (T) to signify that the file is written in conformance with the new extension standard. The presence of the keyword does *not* imply that any conforming extension records are actually appended to the file but merely that they *might* be. Note that EXTEND=T may be used even if random-groups and prototype tables extension records are present so long as their order conforms to the rules specified in the next section. An example of a minimal FITS header with a data matrix is :

```
0........1.........2.........3.........
123456789012345678901234567890123456789
SIMPLE   =                    T /
BITPIX   =                   16 /
NAXIS    =                    2 /
NAXIS1   =                  320 /
NAXIS2   =                  512 /
EXTEND   =                    T /
END
```

An example of a minimal FITS header *without* a data matrix is :

```
0........1.........2.........3.........
123456789012345678901234567890123456789
SIMPLE   =                    T /
BITPIX   =                    8 /
NAXIS    =                    0 /
EXTEND   =                    T /
END
```

Note that the presence of EXTEND=T in a primary FITS header merely indicates that the file *may* have extensions records and that any special records will conform to the rules given below.

7. Structure of files including extension records.

The solution to the problem of compatibility with previously existing extension formats is to specify three new rules :

— If NAXIS1=0 and the random groups keywords are present in the primary header then the random-groups data records (see Greisen and Harten, 1981) are present and they come immediately after the primary header.

— Extension records of the new type must follow the primary matrix or random groups records. Each extension should begin with a FITS-like header which is described below. This header specifies the « type » for the extension and a length which is computed by the usual FITS rules. The header may be arbitrarily long and is terminated by END in the same manner as a primary FITS header. Following the extension header, the specified number of records will appear containing the extension information. The next extension follows immediately after the previous one. Each new extension header must begin with a new record. As many extensions may be included in a FITS file as are required.

— Records of any nonstandard extensions should appear at the end of the file. A reading program should be prepared to encounter such records in any position which would ordinarily contain the first record of a standard extension header and when it does it can assume that the remaining special records of the file are non-standard. The program should examine the first eight bytes of the first record of the putative extension header. If they are the string XTENSION, then the record is the beginning of a standard extension header. If they are not, then it is the first of the non-standard special records.

8. The extension header.

Each extension will begin at the start of a new record (2880 bytes per record). It will contain a standard FITS header except that the first line of the header (normally SIMPLE=T) is replaced by the new keyword XTENSION='type' in order to identify the type of the extension. The required FITS keywords BITPIX, NAXIS, and NAXISnnn are used to specify the dimensions of the binary data matrix of the extension data. Only the standard FITS data types will be acceptable for data interchange. Although the new extension standard allows other values of BITPIX to be used for special purposes, these will be considered to be nonstandard usage.

In order to permit random-groups data structures to be written in the new extensions without the inelegant NAXIS1=0 convention, which had to be adopted for the original random-groups format, the new rules require that *all* extension headers must incorporate the PCOUNT and GCOUNT keywords. For simple matrices rices their value should be PCOUNT=0 and GCOUNT=1. The number of bits which the extension data will occupy will be computed with the following formula :

$$\text{NBITS} = \text{ABS(BITPIX)} * \text{GCOUNT} * (\text{PCOUNT} + \text{NAXIS1} * \text{NAXIS2} * \cdots * \text{NAXIS}n)$$

If NAXIS=0 then the NAXIS*i* terms in the above formula are all zero. Note that the formula specifies use of the *absolute value* of BITPIX ; this is in anticipation of international agreement on a floating point format, in which (IEEE) floating point pixel values will be signified by a negative BITPIX. Also note that the calculation specified here may cause an integer overflow if it is performed with insufficient precision (16 bits rather than 32, for example). The number of standard FITS records (2880 bytes, 23040 bits) which the data will occupy will be computed with this formula :

$$\text{NRECORDS} = \text{INT}\ ((\text{NBITS} + 23039)/23040)$$

Please note that these calculations will apply to all extensions regardless of the type of data structure. This permits designers to utilize BITPIX, GCOUNT, PCOUNT, and the NAXISn in any way which seems appropriate to define their data structures, subject to the constraint that the number of bits computed by the formula above must be correct. Each extension header will be terminated with an END statement in the usual fashion.

The inclusion of GCOUNT and PCOUNT in the extensions allows users considerable flexibility in designing extensions for data which has a semi-regular structure, i.e., the information table or data has a regular size, but there are a few values which are associated with each sub-set of the information and these vary in value with each group. The power of this option is discussed in the first FITS extension paper by Greisen and Harten (1981).

Implementors should note that the extension mechanism should not be used to transmit a 3-dimensional matrix as a sequence of 2-dimensional matrices in separate extension records. Instead, the generalized tools of FITS, in this case the ability to transmit n-dimensional matrices in the Basic FITS header and data matrix, should be used. The freedom provided in the new extension design does not remove from implemen- the obligation to use good taste and standard conventions in their designs. Extensions are meant to be used for other kinds of data such as tables, lists, text files, etc. Implementors of FITS writing programs should always be aware of the limitations of recipients. The primary objective of the FITS standard remains the reliable, unambiguous transmission of data to recipients. Esoteric, complex data structures should be avoided as much as possible. The watchword of the implementor should be : keep it simple.

A typical extension header with no associated data records is :

```
0........1.........2.........3.........4.........5.........6.........7.....
12345678901234567890123456789012345678901234567890123456789012345678901 2...
XTENSION= ' type    '                / the type of the extension
BITPIX  =                    8 /
NAXIS   =                    0 /
PCOUNT  =                    0 /
GCOUNT  =                    1 /
          .
          .
END
```

In the case shown above the extension information is contained solely inside the extension header itself. A minimal extension header with associated data records is :

```
0........1.........2.........3.........4.........5.........6.........7.....
12345678901234567890123456789012345678901234567890123456789012345678901 2...
XTENSION= 'type    '                 / the type of the extension
BITPIX  =                    8 /
NAXIS   =                    1 /
NAXIS1  =                12345 / number of bytes in the date records
PCOUNT  =                    0 /
GCOUNT  =                    1 /
          .
          .
END.
```

This second example would be accompanied by five data records

$$\text{NRECORDS} = \text{INT}\,((8 * 12345 + 23039)/23040)$$

containing an arbitrary stream of 8-bit unsigned integers 12345 bytes long. Such a one-dimensional matrix could be appropriate for transmitting a text file. The tables extension discussed in the companion paper (Harten *et al.*, 1988) defines a table to be a two-dimensional matrix of 8-bit bytes which is used to convey printable ASCII text.

9. Three new optional keywords.

Three new optional keywords are defined for the extension standard :

— EXTNAME='name'

— EXTVER=n

These two keywords are provided for use in the new extension headers to give unique names and version numbers to individual extensions. This means that a FITS file might contain, for example, three different tables extensions (XTENSION='TABLE'). The first might be called EXTNAME='BS83' with EXTVER=1, the second might also be EXTNAME='BS83' but with EXTVER=3, and the third might be EXTNAME='AGK3' with EXTVER=83. I.e., more than one extension of the same type and same name might occur and would be distinguished by unique version numbers, and version numbers need not start with one or be consecutive. If EXTVER is not specified a default value of one should be assumed by a reading program.

The name can also be used to establish a relational type of data base in the same manner as sub-directory names in some file systems. In this case the relationship between the different extensions is established directly. Names such as « map1.cleancomp1 » or « N1234.field2.starlist » can be used to establish easy to understand relationships between different extensions and even between extensions in different FITS files.

— EXTLEVEL=n

This keyword specifies the level of the current extension header in a hierarchical structure of extension headers. The first level of extension headers has the value set to one. Any level-two headers are subordinate to the last previous level-one header, and any level-three headers are subordinate to the last previous level-two header, etc. This concept permits the transmision of arbitrary hierarchical data structures and file systems in FITS. The authors recommend that initial implementations of the new extension design all utilize EXTLEVEL=1 exclusively until experimental trials have demonstrated feasibility of this concept. If EXTLEVEL is not specified, a default value of one should be assumed by a reading program. If the recipient data processing system is unable to represent the hierarchical structure and encounters an extension with EXTLEVEL greater than one, it should act as though EXTLEVEL is one.

10. Extension data records.

The standard FITS philosophy is to keep headers and data in separate records. The new extension format adheres to this rule even though it wastes space (the unused bytes at the end of the last header record). Therefore the extension data begins in the first byte of the next record after the record containing the END of the extension header.

11. Conclusions.

The introduction of blocking factors larger than one significantly increases the storage efficiency on magnetic tapes. Problems of incompatibility with the original FITS standard have been eased by the introduction of a new standard keyword BLOCKED which can be used to indicate the use of blocked FITS tapes.

The rules for the generalized extension of FITS offer a well defined way in which new types of data structures can be added to the FITS standard. They ensure that extensions can be located and decoded by standard routines without interfering with each other. The generalized extensions add a new dimension of flexibility to the Flexible Image Transport System.

References

HARTEN, R. H., GROSBØL, P., GREISEN, E. W., WELLS, D. C. : 1988, *Astron. Astrophys. Suppl. Ser.* (present issue).
GREISEN, E. W., HARTEN, R. H. : 1981, *Astron. Astrophys. Suppl. Ser.* **44,** 371.
IAU Inf. Bull. No. **49,** 14, 1983.
WELLS, D. C., GREISEN, E. W., HARTEN, R. H. : 1981, *Astron. Astrophys. Suppl. Ser.* **44,** 363.