OGIP Memo OGIP/93-001

Specification of Physical Units within OGIP FITS files

Ian M George & Lorella Angelini

Code 668,
NASA/GSFC,
Greenbelt,
MD 20771

Version: 1995 May 04

SUMMARY

A list of character strings specifying the basic physical units used within OGIP FITS files is given. Rules and guidelines on the construction of compound units are also outlined.

This version conforms to the OGIP FITS Working Group Recommendation R5.

LOG OF SIGNIFICANT CHANGES

Release Date	Sections Changed	Brief Notes
1993 Jan 13		First (internal) Draft
1993 May 11	All	Major revision to all strings
1993 Aug 04	All	Voted on as OFWG Recommendation R5.
1994 Nov 18	All	Made compatible with `LaTeX2HTML` software
1995 May 04	4	Section Added

1 INTRODUCTION

In order to facilitate the use of FITS datasets by downstream s/w and users, it is clearly highly preferable that a common set of character strings are used throughout all files and software to specify the physical units in which the data are stored. This memo lists those current approved for use within the OGIP and should be **strictly** adhered to[1].

In Section 2 we give a list of character strings specifying the basic physical units allowed within OGIP FITS files, including guidelines on the use of decimal multiples and submultiples of units. Section 3 provides a number of rules and guidelines on the construction of compound units, and a number of examples are given in Section 5. It should be noted that within the framework provided here, the ultimate responsible for choosing sensible and convenient unit strings lies with authors of FITS datasets.

Any comments or suggestions (especially if you have a requirement for units not covered my this memo) should be be e-mailed to the authors (`LHEAVX::GEORGE` and/or `LHEAVX::ANGELINI`).

2 BASIC UNITS

In line with IAU recommendations, SI units should be adopted for all physical quantities with the exception of a small number of special units more convenient for astronomy. The character strings specifying the basic units allowed for use within the OGIP are listed in Tables 1 & 2, along with the allowed prefixes for multiples and submultiples listed in Table 3. Note that all strings are **CASE SENSITIVE**.

When used as the value of the TUNITnnn keyword in the header of a FITS extension, it is recommended that the first character of the string is placed immediately after the opening quote in the value field (*i.e.* placed in column 12 of the 'card image').

2.1 IAU-recommended Units

Table 2.1 lists the 7 base SI units regarded as dimensionally independent and 2 supplementary units for plane and solid angles, along with the 15 derived units (formed by combining base and supplementary units) with unique symbols recognized by the IAU. All units are given for completeness, though several are likely to be of little use to X- and γ-ray astronomy. It can be seen that in the majority of cases, the character string is identical to the IAU-recognized symbol for the quantity[2]. The only exception is electric resistance (with symbol Ω, but string `ohm`). Finally it should be noted that despite IAU recommendations, the measurement of plane angles in radians may often

[1] It should be noted that in response to numerous comments from the FITS community, the guidelines outlined here differ significantly from those given in all version of this memo prior to 1993 May 11

[2] Note that the IAU has ruled that a second of time should **NOT** be abbreviated to sec; the string `sec` should **NOT** therefore be used to specify such a quantity within a FITS file.

be considered inconvenient within astronomy. Thus the use of decimal degrees is also allowed (*e.g.* Table 2.2).

2.2 Additional Astronomical Units

Unfortunately, a number of the IAU-recommended units listed in Table 1 are somewhat inconvenient for some astronomical applications. Thus Table 2 lists the OGIP-recommended strings for such quantities. Again, in most cases the character string is identical to the IAU-recognized symbol for the quantity. The only exceptions are:

- degrees of arc (with symbol °, but string `deg`)
- angstrom (with symbol Å, but string `angstrom` so as not to be confused with ampere, `A`).
- the character string `yr` is preferred over the IAU-recognized symbol (a) since it is the same as the symbol used for this quantity in most astronomical journals, and in common usage.
- `arcsec` & `arcmin` since the symbols of $''$ and $'$ may lead to confusion

Finally the 'crab' was commonly used in the early days of X-ray astronomy, and is represented in Table 2. However this unit (or more commonly the 'millicrab') should be avoided wherever possible as the conversion to more standard specifications of flux density is a function of the spectrum of the source (relative to that of the Crab). Strictly speaking, the same is true regarding the use of stellar 'magnitudes' (`mag`).

2.3 Prefixes for Multiples & Submultiples

The IAU-approved list of prefixes to note decimal multiples and submultiples of units are listed in Table 3, along with the corresponding character strings. It can be seen that the character is the same as the IAU-approved symbol for the prefix with the exception of micro (10^{-6}), which has the symbol μ, but character `u`. However, with the exception of centi (10^{-2}), the OGIP strongly recommends the use of prefixes which are powers of 3.

Thus strings to specify decimal multiples & submultiples of the units given in Table 1 can be formed[3] by preceeding the listed character string by the relevant prefix character from Table 3. Decimal multiples & submultiples of the units listed in Table 2 should **NOT** used, with the **exception** of electron volt (`eV`), jansky (`Jy`), parsec (`pc`), and crab (`Crab`, for which millicrab, `mCrab` ONLY is allowed). Compound prefixes (*e.g.* `ZYerg` for 10^{45} erg) should never be used. The result is regarded as a single unit string when constructing compound units (see Section 3). It should be noted that the use of these standard prefixes greatly increases the case sensitivity of the resultant string.

[3]The only exception is kilogram (`kg`) which represents 10^3 grammes. Multiples & submultiples of mass should therefore therefore adhere to the above rules, but using grammes, with the unit string `g`, as a base (hence 10^9 kilogram denoted by `Tg`, 10^{-15} kilogram by `pg` *etc.*). The use of grammes (`g`) as a unit is of course also allowed.

Table 1: IAU-recommended basic units allowed within OGIP datasets & software

Quantity	Unit String	Meaning	Notes
SI base & supplementary units			
length	`m`	metre	
mass	`kg`	kilogram	
time	`s`	second of time	(incl sidereal time if appropriate)
plane angle	`rad`	radian	(but see text)
solid angle	`sr`	steradian	
temperature	`K`	kelvin	
electric current	`A`	ampere	
amount of substance	`mol`	mole	
luminous intensity	`cd`	candela	
IAU-recognized derived units			
frequency	`Hz`	hertz	s^{-1}
energy	`J`	joule	N m
power	`W`	watt	J s^{-1}
electric potential	`V`	volts	J C^{-1}
force	`N`	newton	kg m s^{-2}
pressure, stress	`Pa`	pascal	N m^{-2}
electric charge	`C`	coulomb	A s
electric resistance	`ohm`	ohm	V A^{-1} (see text)
electric conductance	`S`	siemens	A V^{-1}
electric capacitance	`F`	farad	C V^{-1}
magnetic flux	`Wb`	weber	V s
magnetic flux density	`T`	tesla	Wb m^{-2}
inductance	`H`	henry	Wb A^{-1}
luminous flux	`lm`	lumen	cd sr
illuminance	`lx`	lux	lm m^{-2}

Table 2: Additional units allowed within OGIP datasets & software

Quantity		**Unit String**	**Meaning**	**Notes**
plane angle		`deg`	degree of arc	1.745533×10^{-2} rad
		`arcsec`	second of arc	4.848137×10^{-6} rad
		`arcmin`	minute of arc	2.90888×10^{-4} rad
time		`min`	minute	(incl sidereal time if appropriate)
		`h`	hour	(incl sidereal time if appropriate)
		`d`	day	8.64×10^{4} s
		`yr`	year (Julian)	3.15576×10^{7} s (365.25 d)
energy	†	`eV`	electron volt	1.602177×10^{-19} J
		`erg`	erg	10^{-7} J
length		`angstrom`	angstrom	10^{-10} m
		`AU`	astronomical unit	1.49598×10^{11} m
		`lyr`	light year	9.460530^{15} m
	†	`pc`	parsec	3.0857×10^{16} m
events		`count`	counts	
		`photon`	photons	
flux density	†	`Jy`	jansky	10^{-26} W m^{-2} Hz^{-1}
		`mag`	(stellar) magnitude	(see text)
	†	`Crab`	'crab'	(see text)
magnetic field		`G`	gauss	10^{-4} T
area		`pixel`	(image/detector) pixel	
		`barn`	barn	10^{-28} m^{2}
Miscellaneous 'units'				
		`chan`	(detector) channel	
		`bin`	numerous applications	(including the 1-d analogue of pixel)
		`voxel`	3-d analogue of pixel	
		`byte`	(computer) byte	

† - addition of prefixes for decimal multiples & submultiples are allowed (see Section 2.3)

Table 3: Prefixes for multiples & submultiples allowed within OGIP datasets & software

Submultiple	Prefix	Character	Multiple	Prefix	Character
10^{-1}	deci	`d`	10	deca	`da`
10^{-2}	centi	`c`	10^2	hecto	`h`
10^{-3}	milli	`m`	10^3	kilo	`k`
10^{-6}	micro	`u`	10^6	mega	`M`
10^{-9}	nano	`n`	10^9	giga	`G`
10^{-12}	pico	`p`	10^{12}	tera	`T`
10^{-15}	femto	`f`	10^{15}	peta	`P`
10^{-18}	atto	`a`	10^{18}	exa	`E`
10^{-21}	zepto	`z`	10^{21}	zetta	`Z`
10^{-24}	yocto	`y`	10^{24}	yotta	`Y`

3 RULES FOR CONSTRUCTION OF COMPOUND UNITS

The rules and syntax for the construction of compound units are fairly straightforward, involving standard characters preceeding or following the string specifying each component unit.

Throughout this section, the symbols `str1`, `str2` *etc.* are used to denote strings conforming to the rules given in Section 2 to specify physical units.

3.1 Basic Syntax

A compound unit is considered to be formed by a series of sub-strings of component units & mathematical operations. Each of these sub-strings **must** be separated by at least one space or a mathematical operator (`*` or `/`). In line with the rules given below, all such sub-strings should be considered in a multiplicative sense unless beginning with a slash (`/`), or enclosed within one or more pairs of brackets.

3.1.1 Multiplication

Multiplicative units can be specified either:

- by simply using one or more preceeding spaces
  e.g. `'str1 str2'`
  (The recommended method)
- by the use of a single asterix (`*`) with or without one or more spaces on either side
  e.g. `'str1 * str2'`

(This syntax should never be used preceeding a sub-string which starts with a slash – see Section 3.1.2).

3.1.2 Division

Units which form the denominator of a compound expression can be specified either:

- by using a slash (`/`) with or without one or more spaces on either side
  e.g. `'str1 /str2'` or `'str1 / str2'` or `'str1/str2'`
  If such a syntax is used, it is recommended that no space is included between the slash and the unit string.
- by raising a multiplicative unit to a negative power (see below)

It should be stressed that the slash character only effects the sub-string it immediately precedes. Thus unless brackets are used, subsequent sub-strings which also form part of the denominator of the compound expression must also be preceeded by a slash.
e.g. `'str1 /str2 str3'` is equivalent to `'str1 str3 /str2'`
whilst `'str1 /str2 /str3'` is equivalent to `'str1 /(str2 * str3)'`

3.1.3 Raising to Powers

A unit string raised to the power `y` is specified

- by using two asterixes (`**`) followed by the index enclosed within round brackets and with no preceeding or intervening spaces. *e.g.* `'str1**(y)'` or `'str1**(-y)'`
  However, if `y` is positive, then the brackets need not be included, but a following space is recommended if additional sub-strings follow.

3.1.4 The use of brackets

Any number of pairs of round brackets `()` may be used within the string for a compound unit in order to prevent ambiguities. As described within this section, a number of rules always/often require their use. However, it is suggested that conservative use is made of such pairs of brackets in order to minimise the total length of compound strings. (It should be remembered that a maximum of 68 characters are allowed in the card image of keywords.)

3.2 Avoidance of underflows & overflows

The inclusion of numerical factors within the unit string should generally be avoided (by the use of multiples and/or submultiples of component basic units; Section 2.3).

However, occasionally it may be preferable to include such factors on the grounds of user-friendliness and/or to minimise the risk of computer under- or overflows. In such cases, the numerical factor can simply be considered a basic unit string and all rules & guidelines given in Section 2 apply.

It should be remembered, however, that the use of numerical (scaling) factors within the unit string can result in additional overheads in the parsing/interpretation of the string if the string is used for any purpose other than as a 'label'. The inclusion of numerical factors should therefore be avoided wherever possible, and should never replace the use of the TSCALnnn keyword.

The following additional guidelines are suggested:

- the numerical factor should precede any unit strings
- only powers of 10 are used as numerical factors

A prime example is astronomical luminosities, and examples of this case along with a number of more complex examples is given in Section 4.

3.3 Mathematical Operations & Functions

The character strings denoting mathematical operations are listed in Table 4. It should be noted that the (round) brackets are mandatory in all cases in which they are included in the table (with the exception of raising units to positive powers – see Section 3.1.3).

Table 4: Characters & strings allowed within OGIP datasets to denote mathematical operations

String	Meaning
`str1 * str2`	Multiplication (see also Section 3.1.1)
`str1 /str2`	Division (see also Section 3.1.2)
`str1**(y)`	Raised to the power `y` (see also Section 3.1.3)
`log(str1)`	Common Logarithm (to base 10)
`ln(str1)`	Natural Logarithm
`exp(str1)`	Exponential ($\exp^{str1}$)
`sqrt(str1)`	Square root
`sin(str1)`	Sine
`cos(str1)`	Cosine
`tan(str1)`	Tangent
`asin(str1)`	Arc Sine
`acos(str1)`	Arc Cosine
`atan(str1)`	Arc Tangent
`sinh(str1)`	Hyperbolic Sine
`cosh(str1)`	Hyperbolic Cosine
`tanh(str1)`	Hyperbolic Tangent

4 RULES FOR QUANTITIES WITH NO OR UNKNOWN UNITS

The following rules apply:

- If the physical quantities are dimensionless, then a blank string should be used. (The string `NONE`, used in some *ROSAT* files is deprecated).
- If the units of a physical quantity are unknown, then the (case-sensitive) string `UNKNOWN` should be used.

The interpretation of the case when the keyword used to store the unit character string is missing is considered application-specific. For example, applications which expect the the quantity to be dimensionless can take the lack of the keyword to mean just this. However, by default, general applications should take the lack of such a keyword to be equivalent to `UNKNOWN`.

5 EXAMPLES

For illustration, example strings representing a number of compound units are given below, along with a limited number of alternatives. It should be noted that within the rules given in Section 3, the number of spaces between individual unit sub-strings is left a matter for personal preference.

1. STRING = `'count /s'` (recommended)
   Meaning: counts per second
   Alternatives:
   STRING = `'count/s'`
   STRING = `'count s**(-1)'`
   STRING = `'count  / s'`
   STRING = `'    count    /s    '` (discouraged)
2. STRING = `'/pixel /s'` (recommended)
   Meaning: per pixel per second
   Alternatives:
   STRING = `'/(pixel * s)'`
3. STRING = `'count /m**2 /s /eV'` (recommended)
   Meaning: counts per square metre per second per electron volt
   Alternatives:
   STRING = `'count m**(-2) * s**(-1) * eV**(-1)'`
   STRING = `'count /(m**2 * s * eV)'`
4. STRING = `'erg /pixel /s /GHz'` (recommended)
   Meaning: ergs per pixel per second per gigahertz
   Alternatives:

STRING = `'erg /s /GHz /pixel'`
STRING = `'erg /pixel /(s * GHz)'` (discouraged)

5. STRING = `'keV**2 /yr /angstrom'` (recommended)
Meaning: square kiloelectron volts per year per angstrom
Alternatives:
STRING = `'10**(10) keV**2 /yr /m'` (discouraged)
STRING = `'(10**2 MeV)**2 /yr /m'` (strongly discouraged)

6. STRING = `'10**(46) erg /s'` (user-friendly)
Meaning: 10^{46} ergs per second
Alternatives:
STRING = `'10**46 erg /s'`
STRING = `'10**(39) J /s'`
STRING = `'10**(39) W'` (recommended)
STRING = `'10**(15) YW'`
STRING = `'YJ /fs'` (for the purist)

7. STRING = `'10**(-7) J /cm**2 /MeV'`
Meaning: 10^{-7} joules per square centimetre per megahertz
Alternatives:
STRING = `'10**(-9) J m**(-2) eV**(-1)'`
STRING = `'nJ m**(-2) eV**(-1)'`
STRING = `'nJ /m**2 /eV'` (recommended)

8. STRING = `'sqrt(erg /pixel /s /GHz)'` (recommended)
Meaning: The square root of Example 4
Alternatives:
STRING = `'(erg /pixel /s /GHz)**(0.5)'`
STRING = `'(erg /pixel /s /GHz)**(1/2)'` (discouraged)
STRING = `'erg**(0.5) pixel**(-0.5) s**(-0.5) GHz**(-0.5)'` (discouraged)

9. STRING = `'log(photon /m**2 /s /Hz)'` (recommended)
Meaning: The common log of a photon flux density
Alternatives:
STRING = `'log(   photon /m**2 /s /Hz     )'`
(*and those associated with the unit sub-strings*)

10. STRING = `'sin( /pixel /s)'` (recommended)
Meaning: The sine of Example 2
Alternatives:
None (*except those associated with the individual unit sub-strings*)

11. STRING = `'(count /s)  (/pixel /s)'`
Meaning: Example 1 multiplied by Example 2
Alternatives:
STRING = `'(count /s)  * (/pixel /s)'`

STRING = `'count /pixel /s**2'` (recommended)

12. STRING = `'log(photon /cm**2 /s /Hz) /(sin( /pixel /s))'` (recommended)
    Meaning: Example 9 divided by Example 10
    Alternatives:
    STRING = `'log(photon /cm**2 /s /Hz) (sin( /pixel /s))**(-1)'` (discouraged)

ACKNOWLEDGMENTS

We would like to thank the numerous people who have contributed valuable comments to the various drafts of this memo, and in particular Don Wells, Clive Page, Bill Pence & Jonathan McDowell.