A Convention for preallocating header space for FITS keywords

William Pence

30 April 2014

1 Background

The ASCII header in every FITS HDU (Header Data Unit) consists of 1 or more 2880-byte blocks, each of which can hold 36 80-byte keyword records. When writing a new keyword to the header of a FITS file, if the header is full (i.e., the last header block already contains 36 header records, including the END keyword) then it becomes necessary to insert a new 2880-byte FITS block at the end of the header. This in turn requires that any data in the FITS file following the header be shifted down by 2880 bytes in the file to make room for the inserted block. This rewriting operation can cause significant data processing inefficiencies when dealing with large FITS files.

One way to circumvent this problem is to preallocate enough space in the header when the FITS HDU is created to hold the anticipated number of keywords that may be written during later processing of the FITS file. This document describes a simple convention for creating an arbitrarily large amount of reserve space in the header that can be used when writing new keywords.

2 Convention details

Under this convention, any sequence of one or more completely blank keyword records (consisting of 80 ASCII space characters) immediately preceding the END keyword are interpreted as non-significant scratch space, which can be reused when new keywords are written. The *functional* end of the header is defined as located at the beginning of this scratch space area, which is where each new keyword record is written. In the event that all the scratch space becomes filled with keywords, then the traditional procedure of shifting the END keyword down one space in the header to make room for the new keyword should be followed.

In practice, it is usually most convenient to write the desired number of blank keywords into the header just prior to writing the END keyword itself, before writing any actual data records to the FITS HDU. Software that recognizes this convention should then reuse these blank records when writing new keywords to the header. This eliminates the inefficiencies associates with having to insert a new FITS block into an existing FITS file to make room for more keywords. Even if some of this reserved header space remains unused (note that space for 100 keywords only occupies 8K of disk space), this is usually insignificant when dealing with large FITS files.

It should be noted that if a FITS file is processed by software that does not support this convention, then new keywords may be written at the location of the END keyword (i.e., *after* the blank keyword records). This will make the blank keywords unavailable for future use by software that does support this convention and will create what appears to be a gap of blank header keywords in the header. For this reason it may be safest to use this convention on FITS files that are created and processed within a controlled environment where all the software is known to support this convention