GammaLib - Bug #609

Scaled integer FITS images are not read correctly.

12/03/2012 09:29 PM - Knödlseder Jürgen

Status:	Closed	**Start date:**	12/03/2012
Priority:	Normal	**Due date:**	
Assigned To:	Knödlseder Jürgen	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

The FITS image access routines perform automatic scaling of the values using BSCALE and BZERO during reading. This means that integer data for which BSCALE=1e-6 will be automatically multiplied by 1e-6 during reading, hence if reading in an integer array is done, the values will be truncated.

This invalidates basically the design of the FITS image interface, as actually, a FITS image will always be read using the native interface.

The problem was (temporarily?) solved by using the routine ffgiet in GFits::new_image which returns the equivalent data type that is needed to store the scale values. Here the description of the cfitsio interface:
Get the data type or equivalent data type of the image. The first routine returns the physical data type of the FITS image, as given by the BITPIX keyword, with allowed values of BYTE IMG (8), SHORT IMG (16), LONG IMG (32), LONGLONG IMG (64), FLOAT IMG (-32), and DOUBLE IMG (-64). The second routine is similar, except that if the image pixel values are scaled, with non-default values for the BZERO and BSCALE keywords, then the routine will return the 'equivalent' data type that is needed to store the scaled values. For example, if BITPIX = 16 and BSCALE = 0.1 then the equivalent data type is FLOAT IMG. Similarly if BITPIX = 16, BSCALE = 1, and BZERO = 32768, then the the pixel values span the range of an unsigned short integer and the returned data type will be USHORT IMG.

int fits_get_img_type / ffgidt (fitsfile *fptr, > int *bitpix, int *status)
int fits_get_img_equivtype / ffgiet (fitsfile *fptr, > int *bitpix, int *status)

This means that the GFits class now returns the equivalent data type class, thus when writing the image again, the image will not be written as scaled integer image but as double precision image.

This is okay for reading, but maybe not for having a coherent interface design.

History

#1 - 01/31/2013 04:45 PM - Knödlseder Jürgen

- Status changed from Feedback to Closed