Image Compression

Set image compression parameters

**Functions**

|  |  |
| --- | --- |
| [imgCompress](https://in.mathworks.com/help/matlab/ref/matlab.io.fits.imgcompress.html) | Compress HDU from one file into another |
| [isCompressedImg](https://in.mathworks.com/help/matlab/ref/matlab.io.fits.iscompressedimg.html) | Determine if current image is compressed |
| [setCompressionType](https://in.mathworks.com/help/matlab/ref/matlab.io.fits.setcompressiontype.html) | Set image compression type |
| [setHCompScale](https://in.mathworks.com/help/matlab/ref/matlab.io.fits.sethcompscale.html) | Set scale parameter for HCOMPRESS algorithm |
| [setHCompSmooth](https://in.mathworks.com/help/matlab/ref/matlab.io.fits.sethcompsmooth.html) | Set smoothing for images compressed with HCOMPRESS |
| [setTileDim](https://in.mathworks.com/help/matlab/ref/matlab.io.fits.settiledim.html) | Set tile dimensions |

matlab.io.fits.imgCompress

Compress HDU from one file into another

**Syntax**

imgCompress(infptr,outfptr)

**Description**

imgCompress(infptr,outfptr) initializes the output HDU, copies all the keywords, and loops through the input image, compressing the data and writing the compressed data to the output HDU.

This function corresponds to the fits\_img\_compress function in the CFITSIO library C API.

**Examples**

import matlab.io.\*

infptr = fits.openFile('tst0012.fits');

outfptr = fits.createFile('myfile.fits');

fits.setCompressionType(outfptr,'rice');

fits.imgCompress(infptr,outfptr);

fits.closeFile(infptr);

fits.closeFile(outfptr);

# matlab.io.fits.isCompressedImg

Determine if current image is compressed

## Syntax

TF = isCompressedImg(fptr)

## Description

TF = isCompressedImg(fptr) returns true if the image in the current HDU is compressed.

This function corresponds to the fits\_is\_compressed\_image function in the CFITSIO library C API.

## Examples

import matlab.io.\*

fptr = fits.openFile('tst0012.fits');

bool = fits.isCompressedImg(fptr);

fits.closeFile(fptr);

matlab.io.fits.setCompressionType

Set image compression type

**Syntax**

setCompressionType(fptr,comptype)

**Description**

setCompressionType(fptr,comptype) specifies the image compression algorithm that should be used when writing a FITS image.

Supported values for comptype include:

|  |
| --- |
| 'GZIP' |
| 'GZIP2' |
| 'RICE' |
| 'PLIO' |
| 'HCOMPRESS' |
| 'NOCOMPRESS' |

This function corresponds to the fits\_set\_compression\_type function in the CFITSIO library C API.

**Examples**

import matlab.io.\*

fptr = fits.createFile('myfile.fits');

fits.setCompressionType(fptr,'GZIP2');

fits.createImg(fptr,'long\_img',[256 512]);

data = reshape(1:256\*512,[256 512]);

data = int32(data);

fits.writeImg(fptr,data);

fits.closeFile(fptr);

fitsdisp('myfile.fits','mode','full');

# matlab.io.fits.setHCompScale

Set scale parameter for HCOMPRESS algorithm

## Syntax

setHCompScale(fptr,scale)

## Description

setHCompScale(fptr,scale) sets the scale parameter to be used with the HCOMPRESS compression algorithm. Setting the scale parameter causes the algorithm to operate in lossy mode.

This function corresponds to the fits\_set\_hcomp\_scale function in the CFITSIO library C API.

## Examples

import matlab.io.\*

data = 50\*ones(256,512,'double') + 10 \* rand([256 512]);

fptr = fits.createFile('myfile.fits');

fits.setCompressionType(fptr,'HCOMPRESS\_1');

fits.setHCompScale(fptr,2.5);

fits.createImg(fptr,'double\_img',[256 512]);

fits.writeImg(fptr,data);

fits.closeFile(fptr);

fitsdisp('myfile.fits','mode','full');

# matlab.io.fits.setHCompSmooth

Set smoothing for images compressed with HCOMPRESS

## Syntax

setHCompSmooth(fptr,smooth)

## Description

setHCompSmooth(fptr,smooth) sets the smoothing to be used when compressing an image with the HCOMPRESS algorithm. Setting either the scale or smoothing parameter causes the algorithm to operate in lossy mode.

This function corresponds to the fits\_set\_hcomp\_smooth function in the CFITSIO library C API.

## Examples

import matlab.io.\*

data = int32(50\*ones(256,512,'double') + 10 \* rand([256 512]));

fptr = fits.createFile('myfile.fits');

fits.setCompressionType(fptr,'HCOMPRESS');

fits.setHCompSmooth(fptr,1);

fits.createImg(fptr,'long\_img',[256 512]);

fits.writeImg(fptr,data);

fits.closeFile(fptr);

fitsdisp('myfile.fits','mode','full');

# matlab.io.fits.setTileDim

Set tile dimensions

## Syntax

fits.setTileDim(fptr,tiledims)

## Description

fits.setTileDim(fptr,tiledims) specifies the size of the image compression tiles to be used when creating a compressed image.

This function corresponds to the fits\_set\_tile\_dim function in the CFITSIO library C API.

## Examples

import matlab.io.\*

fptr = fits.createFile('myfile.fits');

fits.setCompressionType(fptr,'RICE\_1');

fits.setTileDim(fptr,[64 128]);

fits.createImg(fptr,'byte\_img',[256 512]);

data = ones(256,512,'uint8');

fits.writeImg(fptr,data);

fits.closeFile(fptr);

fitsdisp('myfile.fits','mode','full');