FabIO: Fable Input/Output

API Documentation

May 25, 2011

Contents

1 Package fabio

1.1 Modules

(Section 22, p. 63)
- **pilatusimage**:
  Authors: Henning O.
  (Section 23, p. 64)
- **pnmimage**:
  Authors: Henning O.
  (Section 24, p. 66)
- **readbytestream**: Reads a bytestream
  (Section 25, p. 68)
- **tifimage**:
  Authors: Henning O.
  (Section 26, p. 69)
- **xsdimage**: Authors: Jérôme Kieffer, ESRF email:jerome.kieffer@esrf.fr
  (Section 27, p. 71)

1.2 Variables

Name	Description
version	**Value:** `'0.0.7'`
__package__	**Value:** `'fabio'`

2 Module fabio.GEimage

2.1 Functions

demo()

2.2 Variables

Name	Description
GE_HEADER_INFO	**Value:** `[('ImageFormat', 10, None), ('VersionOfStandardHeader', 2...`
__package__	**Value:** `'fabio'`

2.3 Class GEimage

object

fabio.fabioimage.fabioimage

fabio.GEimage.GEimage

2.3.1 Methods

read(*self*, *fname*, *frame*=`0`)

Read in header into self.header and the data into self.data

Overrides: fabio.fabioimage.fabioimage.read

write(*self*, *fname*, *force_type*=<`type 'numpy.uint16'`>)

Not yet implemented

Overrides: fabio.fabioimage.fabioimage.write

getframe(*self*, *num*)

Returns a frame as a new fabioimage object

Overrides: fabio.fabioimage.fabioimage.getframe

next(*self*)

Get the next image in a series as a fabio image

Overrides: fabio.fabioimage.fabioimage.next

previous(*self*)

Get the previous image in a series as a fabio image

Overrides: fabio.fabioimage.fabioimage.previous

Inherited from fabio.fabioimage.fabioimage(Section 14.4)

__init__(), add(), getheader(), getmax(), getmean(), getmin(), getstddev(), integrate_area(), make_slice(), readheader(), rebin(), resetvals(), toPIL16(), update_header()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

2.3.2 Properties

Name	Description
Inherited from object	
__class__	

3 Module fabio.GEimage_old

```
Reads the header from a GE a-Si Angio Detector

Authors: Henning O. Sorensen & Erik Knudsen
         Center for Fundamental Research: Metal Structures in Four Dimensions
         Risoe National Laboratory
         Frederiksborgvej 399
         DK-4000 Roskilde
         email:erik.knudsen@risoe.dk

        + Jon Wright, ESRF

        The header information has been taken from the script read_GEaSi_data.py
        by
        Antonino Miceli
        Thu Jan  4 13:46:31 CST 2007
```

3.1 Variables

Name	Description
__package__	**Value:** `'fabio'`

3.2 Class GEimage

object

fabio.fabioimage.fabioimage

fabio.GEimage_old.GEimage

3.2.1 Methods

read(*self*, *fname*)

```
Read in header into self.header and
    the data   into self.data
```

Overrides: fabio.fabioimage.fabioimage.read

Inherited from fabio.fabioimage.fabioimage(Section 14.4)

__init__(), add(), getframe(), getheader(), getmax(), getmean(), getmin(), getstddev(), integrate_area(), make_slice(), next(), previous(), readheader(), rebin(), resetvals(), toPIL16(), update_header(), write()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

3.2.2 Properties

Name	Description
Inherited from object	
__class__	

4 Module fabio.HiPiCimage

Authors: Henning O. Sorensen & Erik Knudsen
Center for Fundamental Research: Metal Structures in Four Dimensions
Risoe National Laboratory
Frederiksborgvej 399
DK-4000 Roskilde
email:erik.knudsen@risoe.dk

+ Jon Wright, ESRF

Information about the file format from Masakatzu Kobayashi is highly appreciated

4.1 Variables

Name	Description
__package__	**Value:** 'fabio'

4.2 Class HiPiCimage

object
fabio.fabioimage.fabioimage
fabio.HiPiCimage.HiPiCimage

Read HiPic images e.g. collected with a Hamamatsu CCD camera

4.2.1 Methods

read(*self*, *fname*)

Read in header into self.header and
the data into self.data

Overrides: fabio.fabioimage.fabioimage.read

Inherited from fabio.fabioimage.fabioimage(Section 14.4)

__init__(), add(), getframe(), getheader(), getmax(), getmean(), getmin(), getstddev(), integrate_area(), make_slice(), next(), previous(), readheader(), rebin(), resetvals(), toPIL16(), update_header(), write()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

4.2.2 Properties

Name	Description
Inherited from object	
__class__	

5 Module fabio.OXDimage

```
Reads Oxford Diffraction Sapphire 3 images

Authors: Henning O. Sorensen & Erik Knudsen
         Center for Fundamental Research: Metal Structures in Four Dimensions
         Risoe National Laboratory
         Frederiksborgvej 399
         DK-4000 Roskilde
         email:erik.knudsen@risoe.dk

        + Jon Wright, ESRF
```

5.1 Variables

Name	Description
DETECTOR_TYPES	**Value:** {0: 'Sapphire/KM4CCD (1x1: 0.06mm, 2x2: 0.12mm)', 1: 'Sap...
__package__	**Value:** 'fabio'

5.2 Class OXDimage

object

fabio.fabioimage.fabioimage

fabio.OXDimage.OXDimage

5.2.1 Methods

read(*self*, *fname*)

```
Read in header into self.header and
    the data   into self.data
```

Overrides: fabio.fabioimage.fabioimage.read

Inherited from fabio.fabioimage.fabioimage(Section 14.4)

__init__(), add(), getframe(), getheader(), getmax(), getmean(), getmin(), getstddev(), integrate_area(), make_slice(), next(), previous(), readheader(), rebin(), resetvals(), toPIL16(), update_header(), write()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

5.2.2 Properties

Name	Description
Inherited from object	
__class__	

6 Module fabio.adscimage

Authors: Henning O. Sorensen & Erik Knudsen
Center for Fundamental Research: Metal Structures in Four Dimensions
Risoe National Laboratory
Frederiksborgvej 399
DK-4000 Roskilde
email:erik.knudsen@risoe.dk

+ mods for fabio by JPW

6.1 Functions

test()
testcase

6.2 Variables

Name	Description
__package__	**Value:** `'fabio'`

6.3 Class adscimage

object
└
fabio.fabioimage.fabioimage
└
fabio.adscimage.adscimage

Read an image in ADSC format (quite similar to edf?)

6.3.1 Methods

read(*self*, *fname*)
read in the file
Overrides: fabio.fabioimage.fabioimage.read

write(*self*, *fname*)

Write adsc format

Overrides: fabio.fabioimage.fabioimage.write

Inherited from fabio.fabioimage.fabioimage(Section 14.4)

__init__(), add(), getframe(), getheader(), getmax(), getmean(), getmin(), getstddev(), integrate_area(), make_slice(), next(), previous(), readheader(), rebin(), resetvals(), toPIL16(), update_header()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

6.3.2 Properties

Name	Description
Inherited from object	
__class__	

7 Module fabio.bruker100image

7.1 Variables

Name	Description
__package__	**Value:** `'fabio'`

7.2 Class bruker100image

object
fabio.fabioimage.fabioimage
fabio.brukerimage.brukerimage
fabio.bruker100image.bruker100image

7.2.1 Methods

toPIL16(*self*, *filename*=`None`)

Convert to Python Imaging Library 16 bit greyscale image

FIXME - this should be handled by the libraries now

Overrides: fabio.fabioimage.fabioimage.toPIL16 extit(inherited documentation)

read(*self*, *fname*)

Read in and unpack the pixels (including overflow table

Overrides: fabio.fabioimage.fabioimage.read extit(inherited documentation)

Inherited from fabio.brukerimage.brukerimage(Section 8.3)

write(), write2()

Inherited from fabio.fabioimage.fabioimage(Section 14.4)

__init__(), add(), getframe(), getheader(), getmax(), getmean(), getmin(), getstddev(), integrate_area(), make_slice(), next(), previous(), readheader(), rebin(), resetvals(), update_header()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

7.2.2 Properties

Name	Description
Inherited from object	
__class__	

7.2.3 Class Variables

Name	Description
Inherited from fabio.brukerimage.brukerimage (Section 8.3)	
__headerstring__	

8 Module fabio.brukerimage

Authors: Henning O. Sorensen & Erik Knudsen
Center for Fundamental Research: Metal Structures in Four Dimensions
Risoe National Laboratory
Frederiksborgvej 399
DK-4000 Roskilde
email:erik.knudsen@risoe.dk

Based on: openbruker,readbruker, readbrukerheader functions in the opendata module of ImageD11 written by Jon Wright, ESRF, Grenoble, France

8.1 Functions

test()

a testcase

8.2 Variables

Name	Description
__package__	**Value:** 'fabio'

8.3 Class brukerimage

object

fabio.fabioimage.fabioimage

fabio.brukerimage.brukerimage

Known Subclasses: fabio.bruker100image.bruker100image

Read and eventually write ID11 bruker (eg smart6500) images

8.3.1 Methods

read(*self*, *fname*)

Read in and unpack the pixels (including overflow table

Overrides: fabio.fabioimage.fabioimage.read

write(*self*, *fname*)

```
Writes the image as EDF
FIXME - this should call edfimage.write if that is wanted?
eg:     obj = edfimage(data = self.data, header = self.header)
        obj.write(fname)
        or maybe something like: edfimage.write(self, fname)
```

Overrides: fabio.fabioimage.fabioimage.write

write2(*self*, *fname*)

FIXME: what is this?

Inherited from fabio.fabioimage.fabioimage(Section 14.4)

__init__(), add(), getframe(), getheader(), getmax(), getmean(), getmin(), getstddev(), integrate_area(), make_slice(), next(), previous(), readheader(), rebin(), resetvals(), toPIL16(), update_header()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

8.3.2 Properties

Name	Description
Inherited from object	
__class__	

8.3.3 Class Variables

Name	Description
__headerstring__	**Value:** ''

9 Module fabio.cbfimage

```
Authors: Jérôme Kieffer, ESRF
         email:jerome.kieffer@esrf.fr

Cif Binary Files images are 2D images written by the Pilatus detector and others.
They use a modified (simplified) byte-offset algorithm.

CIF is a library for manipulating Crystallographic information files and tries
to conform to the specification of the IUCR
```

Author: J\xc3\xa9r\xc3\xb4me Kieffer

Contact: jerome.kieffer@esrf.eu

Copyright: European Synchrotron Radiation Facility, Grenoble, France

License: GPLv3+

9.1 Variables

Name	Description
DATA_TYPES	**Value:** {'signed 16-bit integer': <type 'numpy.int16'>, 'signed 3...
MINIMUM_KEYS	**Value:** ['X-Binary-Size-Fastest-Dimension', 'ByteOrder', 'Data ty...
DEFAULT_VALUES	**Value:** {'Data type': 'signed 32-bit integer', 'X-Binary-Element-...
__package__	**Value:** 'fabio'

9.2 Class cbfimage

object

fabio.fabioimage.fabioimage

fabio.cbfimage.cbfimage

Read the Cif Binary File data format

9.2.1 Methods

__init__(*self*, *fname*=`None`)

Constructor of the class CIF Binary File reader.

Parameters

`fname:` the name of the file to open

(type=string)

Overrides: object.__init__

read(*self*, *fname*)

```
Read in header into self.header and
    the data   into self.data
```

Overrides: fabio.fabioimage.fabioimage.read

analysePython(*stream*, *size*)

Analyze a stream of char with any length of exception (2,4, or 8 bytes integers)

Parameters

`stream:` string representing the compressed data

`size:` the size of the output array (of longInts)

Return Value

data decompressed

(type=numpy.ndarrays)

analyseWeave(*stream*, *size*)

Analyze a stream of char with any length of exception (2,4, or 8 bytes integers)

Return Value

data decompressed

(type=numpy.ndarray)

analyseNumpy(*stream*, *size*=None)

```
Analyze a stream of char with any length of exception:
            2, 4, or 8 bytes integers

@return: data decompressed
@rtype: numpy.ndarrays
```

Inherited from fabio.fabioimage.fabioimage(Section 14.4)

add(), getframe(), getheader(), getmax(), getmean(), getmin(), getstddev(), integrate_area(), make_slice(), next(), previous(), readheader(), rebin(), resetvals(), toPIL16(), update_header(), write()

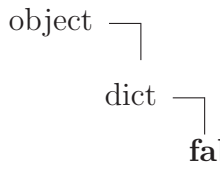
Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

9.2.2 Properties

Name	Description
Inherited from object	
__class__	

9.3 Class CIF



This is the CIF class, it represents the CIF dictionary; and as a a python dictionary thus inherits from the dict built in class.

9.3.1 Methods

__init__(*self*, *_strFilename*=`None`)

Constructor of the class.

Parameters

`_strFilename`: the name of the file to open

(type=filename (str) or file object)

Return Value

`new empty dictionary`

Overrides: object.__init__

readCIF(*self*, *_strFilename*)

Just call loadCIF: Load the CIF file and sets the CIF dictionnary into the object

Parameters

`_strFilename`: the name of the file to open

(type=string)

loadCIF(*self*, *_strFilename*, *_bKeepComment*=`False`)

Load the CIF file and returns the CIF dictionnary into the object

Parameters

`_strFilename`: the name of the file to open

(type=string)

`_bKeepComment`: shall we remove comments

(type=boolean)

Return Value

None

isAscii(*_strIn*)

Check if all characters in a string are ascii,

Parameters

`_strIn`: input string

(type=python string)

Return Value

boolean

(type=boolean)

saveCIF(*self*, *_strFilename='test.cif'*)

Transforms the CIF object in string then write it into the given file

Parameters

`_strFilename`: the of the file to be written

(type=string)

exists(*self*, *sKey*)

Check if the key exists in the CIF and is non empty.

Parameters

`sKey`: CIF key

(type=string)

Return Value

True if the key exists in the CIF dictionary and is non empty

(type=boolean)

existsInLoop(*self*, *sKey*)

Check if the key exists in the CIF dictionary.

Parameters

`sKey`: CIF key

(type=string)

Return Value

True if the key exists in the CIF dictionary and is non empty

(type=boolean)

loadCHIPLOT(*self*, *_strFilename*)

Load the powder diffraction CHIPLOT file and returns the pd_CIF dictionary in the object

Parameters

`_strFilename`: the name of the file to open

(type=string)

Return Value

the CIF object corresponding to the powder diffraction

(type=dictionary)

LoopHasKey(*loop*, *key*)

Returns True if the key (string) exist in the array called loop

Inherited from dict

__cmp__(), __contains__(), __delitem__(), __eq__(), __ge__(), __getattribute__(), __getitem__(), __gt__(), __iter__(), __le__(), __len__(), __lt__(), __ne__(), __new__(), __repr__(), __setitem__(), __sizeof__(), clear(), copy(), fromkeys(), get(), has_key(), items(), iteritems(), iterkeys(), itervalues(), keys(), pop(), popitem(), setdefault(), update(), values()

Inherited from object

__delattr__(), __format__(), __reduce__(), __reduce_ex__(), __setattr__(), __str__(), __subclasshook__()

9.3.2 Properties

Name	Description
Inherited from object	
__class__	

9.3.3 Class Variables

Name	Description
EOL	**Value:** `['\r', '\n', '\r\n', '\n\r']`
BLANK	**Value:** `[' ', '\t', '\r', '\n', '\r\n', '\n\r']`
START_COMMENT	**Value:** `['"', '\'']`
BINARY_MARKER	**Value:** `'--CIF-BINARY-FORMAT-SECTION--'`
Inherited from dict	
__hash__	

10 Module fabio.datIO

```
Authors: Henning O. Sorensen & Erik Knudsen
         Center for Fundamental Research: Metal Structures in Four Dimensions
         Risoe National Laboratory
         Frederiksborgvej 399
         DK-4000 Roskilde
         email:erik.knudsen@risoe.dk

         and Jon Wright, ESRF
```

10.1 Variables

Name	Description
__package__	**Value:** `None`

10.2 Class fabiodata

object
└ **fabio.datIO.fabiodata**

Known Subclasses: fabio.datIO.columnfile

A common class for dataIO in fable Contains a 2d numpy array for keeping data, and two lists (clabels and rlabels) containing labels for columns and rows respectively

10.2.1 Methods

__init__(*self*, *data*=`None`, *clabels*=`None`, *rlabels*=`None`, *fname*=`None`)

set up initial values

Overrides: object.__init__

read(*self*, *fname*=`None`)

To be overridden by format specific subclasses

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),

__repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

10.2.2 Properties

Name	Description
Inherited from object	
__class__	

10.3 Class columnfile

object

fabio.datIO.fabiodata

fabio.datIO.columnfile

10.3.1 Methods

read(*self*, *fname*)

To be overridden by format specific subclasses

Overrides: fabio.datIO.fabiodata.read extit(inherited documentation)

Inherited from fabio.datIO.fabiodata(Section 10.2)

__init__()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

10.3.2 Properties

Name	Description
Inherited from object	
__class__	

11 Module fabio.dm3image

```
Authors: Henning O. Sorensen & Erik Knudsen
         Center for Fundamental Research: Metal Structures in Four Dimensions
         Risoe National Laboratory
         Frederiksborgvej 399
         DK-4000 Roskilde
         email:erik.knudsen@risoe.dk

        + Jon Wright, ESRF
```

11.1 Variables

Name	Description
DATA_TYPES	**Value:** {2: <type 'numpy.int16'>, 3: <type 'numpy.int32'>, 4: <ty...
DATA_BYTES	**Value:** {2: 2, 3: 4, 4: 2, 5: 4, 6: 4, 7: 8, 8: 1, 9: None, 10: N...
__package__	**Value:** 'fabio'

11.2 Class dm3image

object
fabio.fabioimage.fabioimage
fabio.dm3image.dm3image

Read and try to write the dm3 data format

11.2.1 Methods

read(*self*, *fname*)

To be overridden - fill in self.header and self.data

Overrides: fabio.fabioimage.fabioimage.read extit(inherited documentation)

readbytes(*self*, *bytes_to_read*, *format*, *swap*=True)

read_tag_group(*self*)

read_tag_entry(*self*)

read_tag_type(*self*)

read_data(*self*)

Inherited from fabio.fabioimage.fabioimage(Section 14.4)

__init__(), add(), getframe(), getheader(), getmax(), getmean(), getmin(), getstddev(), integrate_area(), make_slice(), next(), previous(), readheader(), rebin(), resetvals(), toPIL16(), update_header(), write()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

11.2.2 Properties

Name	Description
Inherited from object	
__class__	

12 Module fabio.edfimage

```
License: GPLv2+

Authors: Henning O. Sorensen & Erik Knudsen
         Center for Fundamental Research: Metal Structures in Four Dimensions
         Risoe National Laboratory
         Frederiksborgvej 399
         DK-4000 Roskilde
         email:erik.knudsen@risoe.dk

        + Jon Wright, ESRF

2011-02-11: Mostly rewritten by Jérôme Kieffer (Jerome.Kieffer@esrf.eu)
            European Synchrotron Radiation Facility
            Grenoble (France)
```

12.1 Variables

Name	Description
BLOCKSIZE	**Value:** 512
DATA_TYPES	**Value:** {'Double': <type 'numpy.float64'>, 'DoubleIEEE64': <type ...
NUMPY_EDF_DTYPE	**Value:** {'float32': 'FloatValue', 'float64': 'DoubleValue', 'int1...
MINIMUM_KEYS	**Value:** ['HEADERID', 'IMAGE', 'BYTEORDER', 'DATATYPE', 'DIM_1', '...
DEFAULT_VALUES	**Value:** {}
__package__	**Value:** 'fabio'

12.2 Class Frame

object
└ **fabio.edfimage.Frame**

A class representing a single frame in an EDF file

12.2.1 Methods

__init__(*self*, *data*=`None`, *header*=`None`, *header_keys*=`None`, *number*=`None`)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

parseheader(*self*, *block*)

Parse the header in some EDF format from an already open file

Parameters

`block:` string representing the header block

(type=string, should be full ascii)

Return Value

size of the binary blob

swap_needed(*self*)

Decide if we need to byteswap

getData(*self*)

Unpack a binary blob according to the specification given in the header

Return Value

dataset as numpy.ndarray

setData(*self*, *npa*=`None`)

Setter for data in edf frame

getEdfBlock(*self*, *force_type*=`None`)

Parameters

`force_type:` type of the dataset to be enforced like "float64" or "uint16"

(type=string or numpy.dtype)

Return Value

ascii header block

(type=python string with the concatenation of the ascii header and the binary data block)

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

12.2.2 Properties

Name	Description
data	Unpack a binary blob according to the specification given in the header
Inherited from object __class__	

12.3 Class edfimage

object

fabio.fabioimage.fabioimage

fabio.edfimage.edfimage

Read and try to write the ESRF edf data format

12.3.1 Methods

__init__(*self*, *data*=None, *header*=None, *header_keys*=None, *frames*=None)

Set up initial values

Overrides: object.__init__ extit(inherited documentation)

read(*self*, *fname*)

```
Read in header into self.header and
    the data   into self.data
```

Overrides: fabio.fabioimage.fabioimage.read

swap_needed(*self*)

Decide if we need to byteswap

unpack(*self*)

Unpack a binary blob according to the specification given in the header and return the dataset

Return Value
dataset as numpy.ndarray

getframe(*self*, *num*)

returns the file numbered 'num' in the series as a fabioimage

Overrides: fabio.fabioimage.fabioimage.getframe

previous(*self*)

returns the previous file in the series as a fabioimage

Overrides: fabio.fabioimage.fabioimage.previous

next(*self*)

returns the next file in the series as a fabioimage

Overrides: fabio.fabioimage.fabioimage.next

write(*self*, *fname*, *force_type*=`None`)

Try to write a file check we can write zipped also mimics that fabian was writing uint16 (we sometimes want floats)

Parameters
`force_type:` can be numpy.uint16 or simply "float"

Return Value
None

Overrides: fabio.fabioimage.fabioimage.write

appendFrame(*self*, *frame*=`None`, *data*=`None`, *header*=`None`)

Method used add a frame to an EDF file

Parameters
`frame:` frame to append to edf image

(type=instance of Frame)

Return Value
None

deleteFrame(*self*, *frameNb*=`None`)

Method used to remove a frame from an EDF image. by default the last one is removed.

Parameters

`frameNb`: frame number to remove, by default the last.

(type=integer)

Return Value

None

getNbFrames(*self*)

Getter for number of frames

setNbFrames(*self*, *val*)

Setter for number of frames ... should do nothing. Here just to avoid bugs

getHeader(*self*)

Getter for the headers. used by the property header,

setHeader(*self*, *_dictHeader*)

Enforces the propagation of the header to the list of frames

delHeader(*self*)

Deleter for edf header

getHeaderKeys(*self*)

Getter for edf header_keys

setHeaderKeys(*self*, *_listtHeader*)

Enforces the propagation of the header_keys to the list of frames

Parameters

`_listtHeader`: list of the (ordered) keys in the header

(type=python list)

delHeaderKeys(*self*)

Deleter for edf header_keys

getData(*self*)

getter for edf Data

Return Value
data for current frame
(type=numpy.ndarray)

setData(*self*, *_data*)

Enforces the propagation of the data to the list of frames

Parameters
`_data`: numpy array representing data

delData(*self*)

deleter for edf Data

getCapsHeader(*self*)

getter for edf headers keys in upper case

Return Value
data for current frame
(type=dict)

setCapsHeader(*self*, *_data*)

Enforces the propagation of the header_keys to the list of frames

Parameters
`_data`: numpy array representing data

delCapsHeader(*self*)

deleter for edf capsHeader

getDim1(*self*)

setDim1(*self*, *_iVal*)

getDim2(*self*)

setDim2(*self*, *_iVal*)

getDims(*self*)

getByteCode(*self*)

setByteCode(*self*, *_iVal*)

getBpp(*self*)

setBpp(*self*, *_iVal*)

Inherited from fabio.fabioimage.fabioimage(Section 14.4)

add(), getheader(), getmax(), getmean(), getmin(), getstddev(), integrate_area(), make_slice(), readheader(), rebin(), resetvals(), toPIL16(), update_header()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

12.3.2 Properties

Name	Description
nframes	Getter for number of frames
header	property: header of EDF file
header_keys	property: header_keys of EDF file
data	property: data of EDF file
capsHeader	property: capsHeader of EDF file, i.e. the keys of the header in UPPER case.
dim1	
dim2	
dims	
bytecode	
bpp	
Inherited from object __class__	

13 Module fabio.fabian_mar_header

```
This is the mar header interpreter from Fabian by HOS and EK.
JPW replaced it with a parser based approach because:
   this might be hard to maintain, in the even of a single incorrect number somewhere
   theoretically all header items are picked up by the parser, automatically
   the pylint score for this was rather low
```

13.1 Variables

Name	Description
__package__	**Value:** `None`

13.2 Class obselete

14 Module fabio.fabioimage

Authors: Henning O. Sorensen & Erik Knudsen
Center for Fundamental Research: Metal Structures in Four Dimensions
Risoe National Laboratory
Frederiksborgvej 399
DK-4000 Roskilde
email:erik.knudsen@risoe.dk

and Jon Wright, Jerome Kieffer: ESRF

14.1 Functions

test()
check some basic fabioimage functionality

14.2 Variables

Name	Description
__package__	**Value:** `'fabio'`

14.3 Class fabioStream

StringIO.StringIO
└
fabio.fabioimage.fabioStream

just an interface providing the name anf mode property to a StringIO

BugFix for MacOSX

14.3.1 Methods

__**init**__(*self*, *data*, *fname*=`None`, *mode*=`'r'`)
Overrides: StringIO.StringIO.__init__

Inherited from StringIO.StringIO

__iter__(), close(), flush(), getvalue(), isatty(), next(), read(), readline(), readlines(), seek(), tell(), truncate(), write(), writelines()

14.4 Class fabioimage

object
└
fabio.fabioimage.fabioimage

Known Subclasses: fabio.pnmimage.pnmimage, fabio.tifimage.tifimage, fabio.OXDimage.OXDimage, fabio.fit2dmaskimage.fit2dmaskimage, fabio.edfimage.edfimage, fabio.fit2dspreadsheetimage.fit2dspreadshe fabio.mar345image.mar345image, fabio.kcdimage.kcdimage, fabio.dm3image.dm3image, fabio.GEimage_old fabio.adscimage.adscimage, fabio.cbfimage.cbfimage, fabio.xsdimage.xsdimage, fabio.brukerimage.brukerim fabio.HiPiCimage.HiPiCimage, fabio.GEimage.GEimage

A common object for images in fable Contains a numpy array (.data) and dict of meta data (.header)

14.4.1 Methods

__init__(*self*, *data*=`None`, *header*=`None`)

Set up initial values

Overrides: object.__init__

getframe(*self*, *num*)

returns the file numbered 'num' in the series as a fabioimage

previous(*self*)

returns the previous file in the series as a fabioimage

next(*self*)

returns the next file in the series as a fabioimage

toPIL16(*self*, *filename*=`None`)

Convert to Python Imaging Library 16 bit greyscale image

FIXME - this should be handled by the libraries now

getheader(*self*)

returns self.header

getmax(*self*)

Find max value in self.data, caching for the future

getmin(*self*)

Find min value in self.data, caching for the future

make_slice(*self*, *coords*)

Convert a len(4) set of coords into a len(2) tuple (pair) of slice objects the latter are immutable, meaning the roi can be cached

integrate_area(*self*, *coords*)

Sums up a region of interest if len(coords) == 4 -> convert coords to slices if len(coords) == 2 -> use as slices floor -> ? removed as unused in the function.

getmean(*self*)

return the mean

getstddev(*self*)

return the standard deviation

add(*self*, *other*)

Add another Image - warnign, does not clip to 16 bit images by default

resetvals(*self*)

Reset cache - call on changing data

rebin(*self*, *x_rebin_fact*, *y_rebin_fact*, *keep_I*=`True`)

Rebin the data and adjust dims

Parameters

`x_rebin_fact`: x binning factor
(type=int)

`y_rebin_fact`: y binning factor
(type=int)

`keep_I`: shall the signal increase ?
(type=boolean)

write(*self*, *fname*)

To be overwritten - write the file

readheader(*self*, *filename*)

Call the _readheader function...

update_header(*self*, ***kwds*)

update the header entries by default pass in a dict of key, values.

read(*self*, *filename*)

To be overridden - fill in self.header and self.data

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

14.4.2 Properties

Name	Description
Inherited from object	
__class__	

15 Module fabio.fabioutils

15.1 Functions

construct_filename(**args*, ***kwds*)

getnum(*name*)

try to figure out a file number # guess it starts at the back

numstem(*name*)

cant see how to do without reversing strings Match 1 or more digits going backwards from the end of the string

deconstruct_filename(*filename*)

Break up a filename to get image type and number

next_filename(*name*, *padding*=`True`)

increment number

previous_filename(*name*, *padding*=`True`)

decrement number

jump_filename(*name*, *num*, *padding*=`True`)

jump to number

extract_filenumber(*name*)

extract file number

15.2 Variables

Name	Description
FILETYPES	**Value:** {'cbf': ['cbf'], 'cbf.bz2': ['cbf'], 'cbf.gz': ['cbf'], '...
COMPRESSORS	**Value:** {'.bz2': 'bzip2 -dc ', '.gz': 'gzip -dc '}
lines	**Value:** 'bzip2, a block-sorting file compressor. Version 1.0.5, ...
__package__	**Value:** 'fabio'

continued on next page

Name	Description
key	**Value:** `'cbf'`

15.3 Class filename_object

The 'meaning' of a filename

15.3.1 Methods

__init__(*self*, *stem*, *num*=`None`, *directory*=`None`, *format*=`None`, *extension*=`None`, *postnum*=`None`, *digits*=`4`)

str(*self*)

Return a string representation

tostring(*self*)

convert yourself to a string

16 Module fabio.file_series

Authors: Henning O. Sorensen & Erik Knudsen
Center for Fundamental Research: Metal Structures in Four Dimensions
Risoe National Laboratory
Frederiksborgvej 399
DK-4000 Roskilde
email:erik.knudsen@risoe.dk

+ Jon Wright, ESRF

16.1 Functions

new_file_series0(*first_object*, *first*=None, *last*=None, *step*=1)

Created from a fabio image first and last are file numbers

new_file_series(*first_object*, *nimages*=0, *step*=1, *traceback*=False)

```
A generator function that creates a file series starting from a a fabioimage.
Iterates through all images in a file (if more than 1), then proceeds to
the next file as determined by fabio.next_filename.

first_object: the starting fabioimage, which will be the first one yielded
  in the sequence
nimages:  the maximum number of images to consider
step: step size, will yield the first and every step'th image until nimages
  is reached.  (e.g. nimages = 5, step = 2 will yield 3 images (0, 2, 4)
traceback: if True causes it to print a traceback in the event of an
  exception (missing image, etc.).  Otherwise the calling routine can handle
  the exception as it chooses
yields: the next fabioimage in the series.
  In the event there is an exception, it yields the sys.exec_info for the
  exception instead.  sys.exec_info is a tuple:
    ( exceptionType, exceptionValue, exceptionTraceback )
  from which all the exception information can be obtained.
  Suggested usage:
    for obj in new_file_series( ... ):
      if not isinstance( obj, fabio.fabioimage.fabioimage ):
        # deal with errors like missing images, non readable files, etc
        # e.g.
        traceback.print_exception(obj[0], obj[1], obj[2])
```

16.2 Variables

Name	Description
__package__	**Value:** 'fabio'

16.3 Class file_series

object
  list
    fabio.file_series.file_series

Known Subclasses: fabio.file_series.numbered_file_series

```
represents a series of files to iterate
has an idea of a current position to do next and prev

You also get from the list python superclass:
   append
   count
   extend
   insert
   pop
   remove
   reverse
   sort
```

16.3.1 Methods

__init__(*self*, *list_of_strings*)

arg should be a list of strings which are filenames

Return Value
new empty list

Overrides: object.__init__

first(*self*)

first image in series

last(*self*)

last in series

previous(*self*)

prev in a sequence

current(*self*)

current position in a sequence

next(*self*)

next in a sequence

jump(*self*, *num*)

goto a position in sequence

len(*self*)

number of files

first_image(*self*)

first image in a sequence

last_image(*self*)

last image in a sequence

next_image(*self*)

Return the next image

previous_image(*self*)

Return the previous image

jump_image(*self*, *num*)

jump to and read image

current_image(*self*)

current image in sequence

first_object(*self*)

first image in a sequence

last_object(*self*)

last image in a sequence

next_object(*self*)

Return the next image

previous_object(*self*)

Return the previous image

jump_object(*self*, *num*)

jump to and read image

current_object(*self*)

current image in sequence

Inherited from list

__add__(), __contains__(), __delitem__(), __delslice__(), __eq__(), __ge__(), __getattribute__(), __getitem__(), __getslice__(), __gt__(), __iadd__(), __imul__(), __iter__(), __le__(), __len__(), __lt__(), __mul__(), __ne__(), __new__(), __repr__(), __reversed__(), __rmul__(), __setitem__(), __setslice__(), __sizeof__(), append(), count(), extend(), index(), insert(), pop(), remove(), reverse(), sort()

Inherited from object

__delattr__(), __format__(), __reduce__(), __reduce_ex__(), __setattr__(), __str__(), __subclasshook__()

16.3.2 Properties

Name	Description
Inherited from object	
__class__	

16.3.3 Class Variables

Name	Description
Inherited from list	
__hash__	

16.4 Class numbered_file_series

object
  list
    fabio.file_series.file_series
      fabio.file_series.numbered_file_series

mydata0001.edf = ”mydata” + 0001 + ”.edf” mydata0002.edf = ”mydata” + 0002 + ”.edf” mydata0003.edf = ”mydata” + 0003 + ”.edf”

16.4.1 Methods

__init__(*self*, *stem*, *first*, *last*, *extension*, *digits*=4, *padding*='Y', *step*=1)

```
stem - first part of the name
step - in case of every nth file
padding - possibility for specifying that numbers are not padded
          with zeroes up to digits
```

Return Value

new empty list

Overrides: object.__init__

Inherited from fabio.file_series.file_series(Section 16.3)

current(), current_image(), current_object(), first(), first_image(), first_object(), jump(), jump_image(), jump_object(), last(), last_image(), last_object(), len(), next(), next_image(), next_object(), previous(), previous_image(), previous_object()

Inherited from list

__add__(), __contains__(), __delitem__(), __delslice__(), __eq__(), __ge__(), __getattribute__(), __getitem__(), __getslice__(), __gt__(), __iadd__(), __imul__(), __iter__(), __le__(), __len__(), __lt__(), __mul__(), __ne__(), __new__(), __repr__(), __reversed__(), __rmul__(), __setitem__(), __setslice__(), __sizeof__(), append(), count(), extend(), index(), insert(), pop(), remove(), reverse(), sort()

Inherited from object

__delattr__(), __format__(), __reduce__(), __reduce_ex__(), __setattr__(), __str__(), __subclasshook__()

16.4.2 Properties

Name	Description
Inherited from object	
__class__	

16.4.3 Class Variables

Name	Description
Inherited from list	
__hash__	

16.5 Class filename_series

Much like the others, but created from a string filename

16.5.1 Methods

__init__(*self*, *filename*)

create from a filename (String)

next(*self*)

increment number

previous(*self*)

decrement number

current(*self*)

return current filename string

jump(*self*, *num*)

jump to a specific number

next_image(*self*)

returns the next image as a fabioimage

prev_image(*self*)

returns the previos image as a fabioimage

current_image(*self*)

returns the current image as a fabioimage

jump_image(*self*, *num*)

returns the image number as a fabioimage

next_object(*self*)

returns the next filename as a fabio.filename_object

previous_object(*self*)

returns the previous filename as a fabio.filename_object

current_object(*self*)

returns the current filename as a fabio.filename_object

jump_object(*self*, *num*)

returns the filename num as a fabio.filename_object

17 Module fabio.fit2dmaskimage

Author: Andy Hammersley, ESRF Translation into python/fabio: Jon Wright, ESRF

17.1 Variables

Name	Description
__package__	**Value:** `'fabio'`

17.2 Class fit2dmaskimage

object
└
fabio.fabioimage.fabioimage
└
fabio.fit2dmaskimage.fit2dmaskimage

Read and try to write Andy Hammersley's mask format

17.2.1 Methods

read(*self*, *fname*)

```
Read in header into self.header and
    the data   into self.data
```

Overrides: fabio.fabioimage.fabioimage.read

write(*self*, *fname*)

Try to write a file check we can write zipped also mimics that fabian was writing uint16 (we sometimes want floats)

Overrides: fabio.fabioimage.fabioimage.write

Inherited from fabio.fabioimage.fabioimage(Section 14.4)

__init__(), add(), getframe(), getheader(), getmax(), getmean(), getmin(), getstddev(), integrate_area(), make_slice(), next(), previous(), readheader(), rebin(), resetvals(), toPIL16(), update_header()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

17.2.2 Properties

Name	Description
Inherited from object	
__class__	

18 Module fabio.fit2dspreadsheetimage

```
Read the fit2d ascii image output
        + Jon Wright, ESRF
```

18.1 Variables

Name	Description
__package__	**Value:** `'fabio'`

18.2 Class fit2dspreadsheetimage

object
└
fabio.fabioimage.fabioimage
└
fabio.fit2dspreadsheetimage.fit2dspreadsheetimage

Read a fit2d ascii format

18.2.1 Methods

read(*self*, *fname*)

```
Read in header into self.header and
    the data   into self.data
```

Overrides: fabio.fabioimage.fabioimage.read

Inherited from fabio.fabioimage.fabioimage(Section 14.4)

__init__(), add(), getframe(), getheader(), getmax(), getmean(), getmin(), getstddev(), integrate_area(), make_slice(), next(), previous(), readheader(), rebin(), resetvals(), toPIL16(), update_header(), write()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

18.2.2 Properties

Name	Description
Inherited from object __class__	

19 Module fabio.kcdimage

```
Authors: Jerome Kieffer, ESRF
         email:jerome.kieffer@esrf.fr

kcd images are 2D images written by the old KappaCCD diffractometer built by Nonius in t
Based on the edfimage.py parser.
```

19.1 Variables

Name	Description
DATA_TYPES	**Value:** {'u16': <type 'numpy.uint16'>}
MINIMUM_KEYS	**Value:** ['ByteOrder', 'Data type', 'X dimension', 'Y dimension', ...
DEFAULT_VALUES	**Value:** {'Data type': 'u16'}
__package__	**Value:** 'fabio'

19.2 Class kcdimage

object
└
fabio.fabioimage.fabioimage
└
fabio.kcdimage.kcdimage

Read the Nonius kcd data format

19.2.1 Methods

read(*self*, *fname*)

```
Read in header into self.header and
    the data   into self.data
@param fname: input file name
@type fname: string
```

Overrides: fabio.fabioimage.fabioimage.read

Inherited from fabio.fabioimage.fabioimage(Section 14.4)

__init__(), add(), getframe(), getheader(), getmax(), getmean(), getmin(), getstddev(), integrate_area(), make_slice(), next(), previous(), readheader(), rebin(), resetvals(), toPIL16(), update_header(), write()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

19.2.2 Properties

Name	Description
Inherited from object	
__class__	

20 Module fabio.mar345image

Authors: Henning O. Sorensen & Erik Knudsen
Center for Fundamental Research: Metal Structures in Four Dimensions
Risoe National Laboratory
Frederiksborgvej 399
DK-4000 Roskilde
email:erik.knudsen@risoe.dk
+
Jon Wright, ESRF, France

20.1 Variables

Name	Description
__package__	**Value:** 'fabio'

20.2 Class mar345image

object
fabio.fabioimage.fabioimage
fabio.mar345image.mar345image

20.2.1 Methods

read(*self*, *fname*)

Read a mar345 image

Overrides: fabio.fabioimage.fabioimage.read

write(*self*)

To be overwritten - write the file

Overrides: fabio.fabioimage.fabioimage.write extit(inherited documentation)

Inherited from fabio.fabioimage.fabioimage(Section 14.4)

__init__(), add(), getframe(), getheader(), getmax(), getmean(), getmin(), getstddev(), integrate_area(), make_slice(), next(), previous(), readheader(), rebin(), resetvals(), toPIL16(), update_header()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

20.2.2 Properties

Name	Description
Inherited from object	
__class__	

21 Module fabio.marccdimage

Authors: Henning O. Sorensen & Erik Knudsen
Center for Fundamental Research: Metal Structures in Four Dimensions
Risoe National Laboratory
Frederiksborgvej 399
DK-4000 Roskilde
email:henning.sorensen@risoe.dk

+ (mods for fabio) Jon Wright, ESRF

marccdimage can read MarCCD and MarMosaic images including header info.

JPW : Use a parser in case of typos (sorry?)

21.1 Functions

make_format(*c_def_string*)

Reads the header definition in c and makes the format string to pass to struct.unpack

interpret_header(*header*, *fmt*, *names*)

given a format and header interpret it

21.2 Variables

Name	Description
CDEFINITION	**Value:** '\ntypedef struct frame_header_type {\n /* File/h...
C_TO_STRUCT	**Value:** {'INT32': 'i', 'UINT16': 'H', 'UINT32': 'I', 'char': 'c'}
C_SIZES	**Value:** {'INT32': 4, 'UINT16': 2, 'UINT32': 4, 'char': 1}
MAXIMAGES	**Value:** 9
HEADER_FORMAT	**Value:** 'IccccccccccccccccIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII.
HEADER_NAMES	**Value:** ['header_type', 'header_name[16]', 'header_name[16]', 'he...

continued on next page

Name	Description
__package__	**Value:** `'fabio'`

21.3 Class marccdimage

object
fabio.fabioimage.fabioimage
fabio.tifimage.tifimage
fabio.marccdimage.marccdimage

Read in data in mar ccd format, also MarMosaic images, including header info

21.3.1 Methods

Inherited from fabio.tifimage.tifimage(Section 26.2)

__init__(), read(), write()

Inherited from fabio.fabioimage.fabioimage(Section 14.4)

add(), getframe(), getheader(), getmax(), getmean(), getmin(), getstddev(), integrate_area(), make_slice(), next(), previous(), readheader(), rebin(), resetvals(), toPIL16(), update_header()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

21.3.2 Properties

Name	Description
Inherited from object	
__class__	

22 Module fabio.openimage

```
Authors: Henning O. Sorensen & Erik Knudsen
         Center for Fundamental Research: Metal Structures in Four Dimensions
         Risoe National Laboratory
         Frederiksborgvej 399
         DK-4000 Roskilde
         email:henning.sorensen@risoe.dk

mods for fabio by JPW
```

22.1 Functions

do_magic(*byts*)

Try to interpret the bytes starting the file as a magic number

openimage(*filename*)

Try to open an image

openheader(*filename*)

return only the header

22.2 Variables

Name	Description
MAGIC_NUMBERS	**Value:** `[('FORMAT : 86', 'bruker'), ('MM\x00*', 'tif'), ('...`
__package__	**Value:** `'fabio'`

23 Module fabio.pilatusimage

```
Authors: Henning O. Sorensen & Erik Knudsen
         Center for Fundamental Research: Metal Structures in Four Dimensions
         Risoe National Laboratory
         Frederiksborgvej 399
         DK-4000 Roskilde
         email:henning.sorensen@risoe.dk

         + (mods for fabio) Jon Wright, ESRF
marccdimage can read MarCCD and MarMosaic images including header info.

JPW : Use a parser in case of typos (sorry?)
```

23.1 Variables

Name	Description
__package__	**Value:** `'fabio'`

23.2 Class pilatusimage

object
fabio.fabioimage.fabioimage
fabio.tifimage.tifimage
fabio.pilatusimage.pilatusimage

Read in Pilatus format, also pilatus images, including header info

23.2.1 Methods

Inherited from fabio.tifimage.tifimage(Section 26.2)

__init__(), read(), write()

Inherited from fabio.fabioimage.fabioimage(Section 14.4)

add(), getframe(), getheader(), getmax(), getmean(), getmin(), getstddev(), integrate_area(), make_slice(), next(), previous(), readheader(), rebin(), resetvals(), toPIL16(), update_header()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

23.2.2 Properties

Name	Description
Inherited from object	
__class__	

24 Module fabio.pnmimage

Authors: Henning O. Sorensen & Erik Knudsen
Center for Fundamental Research: Metal Structures in Four Dimensions
Risoe National Laboratory
Frederiksborgvej 399
DK-4000 Roskilde
email:henning.sorensen@risoe.dk

24.1 Variables

Name	Description
SUBFORMATS	**Value:** ['P1', 'P2', 'P3', 'P4', 'P5', 'P6', 'P7']
HEADERITEMS	**Value:** ['SUBFORMAT', 'DIMENSIONS', 'MAXVAL']
P7HEADERITEMS	**Value:** ['WIDTH', 'HEIGHT', 'DEPTH', 'MAXVAL', 'TUPLTYPE', 'ENDHDR']
__package__	**Value:** 'fabio'

24.2 Class pnmimage

object
fabio.fabioimage.fabioimage
fabio.pnmimage.pnmimage

24.2.1 Methods

__init__(*self*)

Set up initial values

Overrides: object.__init__ extit(inherited documentation)

read(*self*, *fname*, *verbose*=0)

To be overridden - fill in self.header and self.data

Overrides: fabio.fabioimage.fabioimage.read extit(inherited documentation)

P1dec(*self*, *buf*, *bytecode*)

P4dec(*self*, *buf*, *bytecode*)

P2dec(*self*, *buf*, *bytecode*)

P5dec(*self*, *buf*, *bytecode*)

P3dec(*self*, *buf*, *bytecode*)

P6dec(*self*, *buf*, *bytecode*)

P7dec(*self*, *buf*, *bytecode*)

write(*filename*)

To be overwritten - write the file

Overrides: fabio.fabioimage.fabioimage.write extit(inherited documentation)

Inherited from fabio.fabioimage.fabioimage(Section 14.4)

add(), getframe(), getheader(), getmax(), getmean(), getmin(), getstddev(), integrate_area(), make_slice(), next(), previous(), readheader(), rebin(), resetvals(), toPIL16(), update_header()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

24.2.2 Properties

Name	Description
Inherited from object	
__class__	

25 Module fabio.readbytestream

```
Reads a bytestream

Authors: Jon Wright     Henning O. Sorensen & Erik Knudsen
         ESRF           Risoe National Laboratory
```

25.1 Functions

readbytestream(*fil*, *offset*, *x*, *y*, *nbytespp*, *datatype*='int', *signed*='n', *swap*='n', *typeout*=<type 'numpy.uint16'>)

Reads in a bytestream from a file (which may be a string indicating a filename, or an already opened file (should be "rb")) offset is the position (in bytes) where the pixel data start nbytespp = number of bytes per pixel type can be int or float (4 bytes pp) or double (8 bytes pp) signed: normally signed data 'y', but 'n' to try to get back the right numbers when unsigned data are converted to signed (python once had no unsigned numeric types.) swap, normally do not bother, but 'y' to swap bytes typeout is the numpy type to output, normally uint16, but more if overflows occurred x and y are the pixel dimensions

TODO : Read in regions of interest

PLEASE LEAVE THE STRANGE INTERFACE ALONE - IT IS USEFUL FOR THE BRUKER FORMAT

25.2 Variables

Name	Description
DATATYPES	**Value:** {('double', 'y', 4): <type 'numpy.float64'>, ('float', 'y...
__package__	**Value:** 'fabio'

26 Module fabio.tifimage

```
Authors: Henning O. Sorensen & Erik Knudsen
         Center for Fundamental Research: Metal Structures in Four Dimensions
         Risoe National Laboratory
         Frederiksborgvej 399
         DK-4000 Roskilde
         email:henning.sorensen@risoe.dk

mods for fabio by JPW
```

26.1 Variables

Name	Description
TIFF_TO_NUMERIC	**Value:** {'I;16': <type 'numpy.int16'>}
__package__	**Value:** 'fabio'

26.2 Class tifimage

object
└
fabio.fabioimage.fabioimage
└
fabio.tifimage.tifimage

Known Subclasses: fabio.marccdimage.marccdimage, fabio.pilatusimage.pilatusimage

Images in TIF format Wraps PIL

26.2.1 Methods

__init__(*self*, **args*, ***kwds*)

Tifimage constructor adds an nbits member attribute

Overrides: object.__init__

read(*self*, *fname*)

The fabian read was reading a PIL image We convert this to a numpy array

Overrides: fabio.fabioimage.fabioimage.read

write(*self*, *fname*)

... at least try ...

Overrides: fabio.fabioimage.fabioimage.write

Inherited from fabio.fabioimage.fabioimage(Section 14.4)

add(), getframe(), getheader(), getmax(), getmean(), getmin(), getstddev(), integrate_area(), make_slice(), next(), previous(), readheader(), rebin(), resetvals(), toPIL16(), update_header()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

26.2.2 Properties

Name	Description
Inherited from object	
__class__	

27 Module fabio.xsdimage

```
Authors: Jérôme Kieffer, ESRF
         email:jerome.kieffer@esrf.fr

XSDimge are XML files containing numpy arrays
```

Author: J\xc3\xa9r\xc3\xb4me Kieffer

Contact: jerome.kieffer@esrf.eu

Copyright: European Synchrotron Radiation Facility, Grenoble, France

License: GPLv3+

27.1 Variables

Name	Description
__package__	**Value:** `'fabio'`

27.2 Class xsdimage

object
fabio.fabioimage.fabioimage
fabio.xsdimage.xsdimage

Read the XSDataImage XML File data format

27.2.1 Methods

__init__(*self*, *data*=`None`, *header*=`None`, *fname*=`None`)

Constructor of the class XSDataImage.

Parameters

`data:` input data
(type=numpy.ndarray)

`header:` input metadata
(type=dict)

`fname:` the name of the file to open
(type=string)

Overrides: object.__init__

read(*self*, *fname*)

To be overridden - fill in self.header and self.data

Overrides: fabio.fabioimage.fabioimage.read

Inherited from fabio.fabioimage.fabioimage(Section 14.4)

add(), getframe(), getheader(), getmax(), getmean(), getmin(), getstddev(), integrate_area(), make_slice(), next(), previous(), readheader(), rebin(), resetvals(), toPIL16(), update_header(), write()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

27.2.2 Properties

Name	Description
Inherited from object	
__class__	

Index