

# Package ‘Rfa’

November 3, 2015

**Version** 4.0.1-1

**Date** 2015-10-30

**Title** Read and work with binary ALADIN files

**Author** Alex Deckmyn <alex.deckmyn@meteo.be>.

**Maintainer** Alex Deckmyn <alex.deckmyn@meteo.be>

**Depends** geogrid (>= 3.2.1)

**Description** Contains functions for reading and writing binary FA files.

**License** This code is internal for the ALADIN & HIRLAM communities

**URL**

## R topics documented:

biper . . . . .	2
ecto . . . . .	2
FAcreate . . . . .	3
FAdec . . . . .	4
FAechkevo . . . . .	5
FAenc . . . . .	6
FAframe . . . . .	8
FAmake.frame . . . . .	9
FAmake.header . . . . .	10
FAmake.time . . . . .	10
FAopen . . . . .	11
FAprsure . . . . .	12
FAslice . . . . .	13
FAsounding . . . . .	14
internal . . . . .	15
LFIopen . . . . .	16
LFIreplace . . . . .	17
utilities . . . . .	18
<b>Index</b>	<b>19</b>

---

biper	<i>Biperiodic continuation of a field.</i>
-------	--

---

### Description

This is essentially a wrapper to the ETIBIHI routine for biperiodicisation of a 2D field using isotropic splines.

### Usage

```
biper(data,newdim=dim(data)+11)
```

### Arguments

data	A geofield object or a data matrix .
newdim	The dimensions of the final field. By default, the extension zone is 11 points wide.

---

ecto	<i>Spectral energy of a field.</i>
------	------------------------------------

---

### Description

This function implements the basic behaviour of the *ectoplasm* program. It calculates the spectral energy as a function of wave number, for a given field.

### Usage

```
ecto(data,NMSMAX=maxM,NSMAX=maxN)
```

### Arguments

data	A geofield object or a data matrix .
NMSMAX,NSMAX	Highest wave numbers in X and Y direction. If not specified, they are derived from the matrix dimensions of data, which may give a slightly different result.

### Value

A vector of length  $\max(\text{NMSAX}, \text{NSMAX}) + 1$  containing the spectral energy per wave number.

---

FAcreate*Create a new FA file*

---

**Description**

Creates a new FA file based on domain and time information.

**Usage**

```
FAcreate(filename, frame, time=FAmake.time(), sector.size=1000,  
         nsector=10, overwrite=FALSE)
```

**Arguments**

filename	Guess what? A string with the file name.
frame	An FAframe class object, e.g. created by FAmake.frame.
time	A vector of 11 integers representing date and forecast range. Default is to call FAmake.time(), creating a default date.
sector.size	Size of the data sectors (in 8 byte words). Since a field name is 16 bytes, this is double the number of fields that can be encoded without having to do some sector magic.
nsector	How many data sectors of the given size do you want? Absolute minimum is 4. Putting more just gives a larger file to begin with. When you write new fields to the file, data sectors are added on the fly when necessary.
overwrite	Just a simple safety switch.

**Value**

This function creates a new file (with only the 7 meta-information fields) and returns a FAfile class object for the newly created file.

**See Also**

[FAopen](#), [FAmake.frame](#)

FAdec

*Decode an FA field***Description**

Decodes one field of an FA file. Calls different methods depending on the first argument.

**Usage**

```
FAdec(fa,...)
## S3 method for class 'character'
FAdec(fa,field,clip=TRUE,outform="G",archname=NULL,
      tar.offset=NULL,faframe=NULL,fatime=NULL,quiet=TRUE,...)
## S3 method for class 'FAfile'
FAdec(fa,field,clip=TRUE,outform="G",quiet=TRUE,...)
FAmsg.dec(inbuf,faframe,clip=TRUE,outform="G",quiet=TRUE)
```

**Arguments**

fa	A FAfile object as returned by FAopen. fa may also be a character string, in which case it is interpreted as a filename. If multiple fields are decoded from a single file, it is considerably faster to FAopen it only once! For a single field,
field	Can be either an integer, in which case it indicates the position of the field in the fieldlist (the frame is not counted!), or else it can be a character string. field may be any part of the complete name of the field, as long as it is unique. If various names fit the given string, an error will be thrown only if fa is an FAfile object. If fa is a character string (and the file has thus not been parsed completely the first fit will be chosen, even if there are more.
clip	If TRUE, the extension zone is not included in the result.
outform	If "G", the result is a geogrid object. "M" returns the same data matrix but without all the geogrid attributes. This is a bit faster and may be useful when decoding many fields into a larger structure. Other values are "S" for returning the spectral components and "R" for the raw data encoded in the file. These alternatives are only applicable if the field is stored on spectral format, and the result does not include geographical information.
archname, tar.offset	If fa is a file name, it may be part of a tar archive.
faframe, fatime	If fa is a file name, these variables can be added. This speeds up the decoding (since the frame is not read from the file). If faframe is provided, fatime will also not be read from the file, even if it is NULL. The rationale is that we need the frame for decoding the object, but the time info is not vital.
inbuf	A bitstream containing an FA article in binary form.
quiet	Set to TRUE for (a lot of) debugging information.
...	Not used.

## Details

FAdec is a function with two different methods for FAfile and character objects. Some of the arguments are only accepted if fa is a character string. FAmgs.dec is an internal function, which does the actual decoding of a raw bitstream.

## Value

A geofield object, i.e. a matrix containing numerical data. with following attributes:

domain	A list of all grid and projection parameters, as returned by FAgid.
time	A list of analysis and forecast time parameters, as returned by FATime.
info	A description of the field, as returned by FAdescribe.

## See Also

[FAopen](#), [FAenc](#), [iview](#)

## Examples

```
## Not run:
bel1 <- FAopen('ICMSHABOF+0001')
oro <- FAdec(bel1, 'SPECSURFGEO')

## End(Not run)
```

---

FAechkevo

Read FA echkevo files (experimental)

---

## Description

Echkevo files are also partially FA files (frame), but the other fields are different.

## Usage

```
FAechk.open(filename, lswap=TRUE)
FAechk.read(fe, timestep)
FAechkevo(filename, lswap=TRUE)
```

## Arguments

filename	A character string with complete path.
lswap	Set to FALSE if the echkevo file was PRODUCED by a big-endian HPC. This covers a bug in echkevo.
fe	Output from FAechk.open.
tstep	An integer. Returns the output (all points and fields) for a single timestep.

**Value**

FAechkevo returns a list with various elements:

fa	The output from FAopen. This contains all frame information etc.
info	Again a list, with information about which fields and grid points are in the echkevo output.
docu	The numerical values in the "docu" field. Better use the interpreted version in info.
tstep	Timestep in seconds.
geo	
data	The actual data. This is a 3 dimensional array with dimensions(timesteps, fields, points).

**See Also**

[FAdec](#)

---

FAenc

---

*Encode/remove a field in an existing FA file.*


---

**Description**

Encode or remove one field into an existing FA file opened by FAopen. These functions are quite new and should be used with caution!

**Usage**

```
FAenc(fa,fieldname,data,lspec=NULL,lgrib=NULL,
      nbits=NULL,sptrunc=NULL,sppow=NULL,
      overwrite=TRUE,quiet=TRUE)
FAMsg.enc(data, faframe, lspec=NULL, lgrib=NULL, nbits=NULL,
          sptrunc=NULL, sppow=NULL, quiet=TRUE)
FAadd(fa,fieldname,data,quiet=TRUE,...)
FAremove(fa, field, quiet=TRUE)
```

**Arguments**

fa	A FAfile object as returned by FAopen. If fa is a character string, it is interpreted as a file name and opened using FAopen.
data	The data field to be encoded. If the field dimension are smaller than required by the dataframe, the domain is extended.
fieldname	A name for the new field. It must be at least 5 characters long and may not yet exist in the FAfile. If the name already exists, the field is replaced by the new values (unless overwrite==FALSE).
field	Name or number of the field to be removed.

lspec	FALSE stores the data in gridpoint format, TRUE in spectral format.
lgrib	Grib encoding.
nbits	Number of bits per value in the GRIB encoding.
sptrunc	Spectral truncation.
sppow	Power of the laplacian used for rescaling.
overwrite	If FALSE, the function exits if the field name already exists in the given FA file. If overwrite is TRUE an existing field with the same name will be overwritten.
faframe	An FAframe.
quiet	FALSE turns on a lot of diagnostic messages.
...	Options to FAens.msg

## Details

FAenc is the main function for encoding data into FA format, and probably the only one you should use. The other functions are rather internal functions that may be useful in special cases.

To write a data matrix, it must have the correct dimensions. This often means that a biperiodicisation zone must be added. This can be done using the function `biper`. If the data matrix doesn't have a extension zone yet, FAenc will call `biper` automatically.

If the file is not large enough to contain the new data, it is extended. For this reason, these routines will not accept files that are part of a tar archive.

## Value

All these functions return a modified FAfile object resulting from opening the modified file. See the example for typical way of use.

## See Also

[FAopen](#), [FAdec](#), [biper](#)

## Examples

```
## Not run:
fa1 <- FAopen("someFAfile")
fa1 <- FAremove(fa1,"someField")

## End(Not run)
```

---

FAframe	<i>FA meta data ('frame')</i>
---------	-------------------------------

---

## Description

Functions to decode the 'meta-data' encoded in the 7 'frame' articles.

## Usage

```
FAread.meta(filename, archname=NULL, quiet=TRUE)
FAframe(framelist)
FAtime(timelist)
FAdomain(faframe, quiet=TRUE)
```

## Arguments

filename	Guess what.
archname	The file may be part of a tar archive.
framelist	A list containing the 7 'frame' ('cadre') articles as numerical vectors.
timelist	A vector of 11 integer values. This is in fact the 7th vector in the framelist.
faframe	A FAframe class object as returned e.g. by FAframe.
quiet	Set to FALSE for (a lot of) diagnostic information.

## Value

A geofield object, i.e. a matrix containing numerical data. with following attributes:

domain	A list of all grid and projection parameters, as returned by FAgid.
time	A list of analysis and forecast time parameters, as returned by FATime.
info	A description of the field, as returned by FAdescribe.

## See Also

[FAopen](#)



---

FMake.frame	Create a new faframe class object
-------------	-----------------------------------

---

## Description

Encode or remove one field into an existing FA file opened by FAopen. These functions are quite new and should be used with caution!

## Usage

```
FMake.frame(domain,extension=c(11,11),relaxation=c(8,8),nsmax=NULL,nmsmax=NULL,
            lineargrid=TRUE,levels=NULL,
            name="FA-FRAME", sptrunc=10)
```

## Arguments

domain	A geodomain or having a domain as attribute.
extension	Dimension of the extension zone in X and Y directions.
relaxation	Size of teh relaxation zone for Davis coupling. NOT USED?
nmsmax, nsmax	Spectral truncation. Default value depends on the choice of linergrid.
lineargrid	If TRUE, the default values of nmsmax and nsmax are given by floor((nx-1)/2). If FALSE, quadtratic grid is taken: floor((nx-1)/3). If nsmax and nmsmax are given explicetely, this option is ignored.
levels	Hybrid level definition. Default is 1 level.
name	Name for the frame as written in the FA file.
sptrunc	The default wavelength for truncating the GRIB compacted part. Best to leave this at 10.

## Value

Returns a FAframe class object. Only 'aladin' type.

## See Also

[Fcreate](#), [Fmake.header](#), [Fmake.time](#)

---

FMake.header	<i>Create a basic header (22 reals) for a FA file</i>
--------------	---

---

### Description

Returns a basic header for a FAfile. The only option is the sector size.

### Usage

```
FMake.header(sector.size=1000)
```

### Arguments

sector.size	The size of the data sectors for a FA file in 16bit words.
-------------	--

### Value

A header is a vector of 22 reals.

### See Also

[FCreate](#), [FMake.frame](#), [FMake.time](#)

---

FMake.time	<i>Create a list of time entries for a FA file.</i>
------------	---

---

### Description

Encode or remove one field into an existing FA file opened by FAopen. These functions are quite new and should be used with caution!

### Usage

```
FMake.time(fcdate="1970010100", fcrange=0, unit='h', init=1)
```

### Arguments

fcdate	Forecast data, a POSIXct class object. A string or simple number YYYYMMDDHH is also possible.
fcrange	Usually forecast range in hours.
unit	Unit used for range. Usually 'h'. But may be 'm' for minutes.
init	Should the time vector be labelled as 'initialised' or not?

### Value

Returns a vector of length 11 for FA file time sector.

**See Also**

[FCreate](#), [FMake.header](#), [FMake.frame](#)

---

FAopen	<i>Open an FA file</i>
--------	------------------------

---

**Description**

Opens an FA file and returns a `FAlist` object. The data itself is not decoded, but domain specifications, date, list of fields etc are returned.

**Usage**

```
FAopen(filename, archname=NULL, tar.offset=NULL, lparse=TRUE, quiet=TRUE)
FAopenTar(archname, lparse=FALSE, quiet=TRUE)
```

**Arguments**

filename	A character string with complete path.
archname	If the FA file is part of a tar archive. A character string with complete path.
tar.offset	Byte location of the FA file within the tar archive. If <code>NULL</code> , the file is searched for.
lparse	If <code>FALSE</code> , the FA file(s) are only searched for all available fields and their bit addresses. If <code>TRUE</code> , the FA file(s) are parsed completely and the main characteristics of every field are also returned (spectral, grib encoding etc.). This is quite fast and can be useful (it is not vital), but may be slow for large tar archives over a slow connection.
quiet	If <code>FALSE</code> , several diagnostic messages are printed.

**Value**

An `FAfile` object is a list with two components:

list	A list of all available fields (excluding the first 7). The columns contain byte address etc. The number of columns depends on the value of <code>lparse</code>
holes	a list of "holes" in the file resulting from rewrites and removals of fields. Only important for encoding routines.

A `FAfile` object also has a list of attributes:

filename	Full name of the file.
tarfile, tar.offset	<code>NULL</code> and 0 for individual files.
nfields, nholes	Number of fields (holes) contained in the file.
frame	All geographical meta data: domain size, projection, vertical levels etc.
time	Time information (analysis and forecast).

`FAopenTar` returns a list of such objects.

**See Also**[FAdec](#)

FApressure

*Calculate the pressure at a hybrid model level of a FA field***Description**

Calculates pressure field at model levels.

**Usage**

```
FApressure(fa,lev,SP = FAdec(fa, "SURFPRESSION  "))
FApressures.local(faframe,surfpressure)
```

**Arguments**

fa	A FAfile object as returned by FAopen. fa may also be a character string, in which case it is interpreted as a filename (and automatically opened using FAopen).
faframe	A FAframe, but it may also be any class that has a FAframe attribute.
lev	An integer indicating the model level ("S00n...") to calculate pressure field.
SP	Surface pressure field. By default it is taken from the FA file. If it is not available there, SP must be given explicitly.
surfpressure	A single numerical value.

**Details**

The pressure calculation uses the formula  $P = A * P_{ref} + B * P_{surf}$  for half-levels. Then an interpolation is done to full levels.

**Value**

FApressure a geofield containing the pressure field at the requested hybrid model level. FApressures.local returns a vector of pressures for all the hybrid levels, based on the given surface pressure.

**See Also**[FAopen](#), [FAdec](#), [iview](#)**Examples**

```
## Not run:
bel1 <- FAopen('ICMSHABOF+0001')
P35 <- FApressure(bel1,35) ### pressure field at hybrid model level 35

## End(Not run)
```

FAslice

*Extract vertical cross section***Description**

Extracts a vertical cross section from a FA file for 1 parameter, along any line of the grid.

**Usage**

```
FAslice(fa,par="TEMPERATURE", axis="X", n=1, type="S", plot=FALSE,
        plot.function=contour, ...)
```

**Arguments**

fa	A FAfile object as returned by FAopen. fa may also be a character string, in which case it is interpreted as a filename (and automatically opened using FAopen).
par	A character string denoting the field to extract. Default is 'TEMPERATURE'.
type	The level type to extract. Can be 'S' for hybrid levels or 'P' for pressure levels.
axis	Can be 'X' or 'Y' and denotes in which direction the slice is taken.
n	The index (along the other axis) where to take the cross section.
plot	If TRUE, the vertical cross-section is plotted.
plot.function, ...	The function that should draw the plot, and any options that can be passed.

**Value**

A matrix with a cross section of the chosen parameter.

**See Also**

[Fapressure](#), [FAsounding](#)

**Examples**

```
## Not run:
bel1 <- FAopen('ICMSHABOF+0001')
P35 <- FAslice(bel1,axis="X",15)
P35 <- FAslice(bel1,axis="X",15, plot=TRUE, plot.function=limage,
               color.palette=irainbow)

## End(Not run)
```

---

FAsounding

---

*Extract pseudo-soundings*


---

## Description

Extracts vertical samples from a FA file for 1 or more points and parameters.

## Usage

```
FAsounding(fa,par="TEMPERATURE",lon,lat,index=NULL, id=NULL, type="S",
           plevels.out=NULL, method="bilin")
```

## Arguments

fa	A FAfile object as returned by FAopen. fa may also be a character string, in which case it is interpreted as a filename (and automatically opened using FAopen).
par	A character string (or a vector of strings) denoting the fields to extract. Default is 'TEMPERATURE'.
type	The level type to extract. Can be 'S' for hybrid levels or 'P' for pressure levels.
lon,lat	Co-ordinates at which to extract the pseudo-sounding. The fields are interpolated to these locations using 'method'.
index	In stead of longitudes and latitudes, one may also give a set of inideces, a matrix with two columns, e.g. cbind(i,j).
id	If multiple points are extracted, they must be distinguished by an ID. This may be the WMO code, but if it is not provided, the default is to either simply number from 1 to N, or use the index values.
method	Interpolation method. May be 'bilin', 'closets' or 'bicubic'.
plevels.out	A vector of pressure levels (in hPa). If it is NULL, the model levels are returned. If it is defined, the pseudo-sounding is interpolated (using splines and logarithm of pressure).

## Value

A data.frame containing columns "model\_level","pressure" and one column for every entry in par. If pressure.levels is given, there is no column "model\_level".

## See Also

[Fapressure](#), [point.interp](#)

**Examples**

```
## Not run:
bel1 <- FAopen('ICMSHABOF+0001')
P35 <- FApessure(bel1,35) ### pressure field at hybrid model level 35

## End(Not run)
```

internal

*Internal routines***Description**

You will probably never want to use these functions. But who knows...

**Usage**

```
FArawreorder(rawdata, nmsmax, nsmax, inv = FALSE)
FAraw2fft(rawdata,nmsmax,nsmax,ndlon,ndgl)
FAfft2raw(fftdata,nmsmax,nsmax,ndlon,ndgl,quiet=TRUE)
FAheader.date(header,quiet=TRUE)
FAheader.rewrite(fa,ff=NULL)
```

**Arguments**

rawdata	A vector of 'raw' (quadruplet) spectral components.
nmsmax, nsmax	Spectral truncations.
fftdata	A matrix of complex spectral components.
ndlon,ndgl	Grid dimensions.
inv	For the inverse transformation, of course.
header	A vector of length 22, the header of a FA/LFI file.
quiet	Set to FALSE for (a lot of) debugging information.
fa	A FAfile class.
ff	An open file connection. If NULL, a new connection is opened.

**Value**

- FArawreorder changes the order of the quadruplets. This is of importance when reading some GRIB files for data assimilation B-matrix calculations, because there the quadruplets are ordered differently!
- FAraw2fft and FAfft2raw convert from a vector of quadruplets to a complex matrix of FFT components and back.
- FAheader.date updates the 'last change' date in the header.

LFIopen

*Decode an LFI field (SURFEX)***Description**

Read and/or decodes fields of a LFI file.

**Usage**

```
LFIopen(filename, quiet=TRUE)
LFIdec(lfi, field, ...)
LFIread(lfi, field, type="numeric", missing=1.0E+20, add.attr=FALSE, quiet=TRUE)
LFItime(lfi)
LFIdomain(lfi, quiet=TRUE)
```

**Arguments**

filename	Guess what...
lfi	An object as returned by LFIopen. lfi may also be a character string, in which case it is interpreted as a filename (and automatically opened using LFIopen).
field	Can be either an integer, in which case it indicates the position of the field in the fieldlist, or else it can be a character string. If field is a string, it may any part of the complete name of the field, as long as it is unique.
type	Expected data type.
missing	Data value that should be interpreted as missing value. This is replaced by NA in the output.
add.attr	If TRUE, extra attributes (gridtype, mcomment) are added to the decoded field.
quiet	FALSE turns on the diagnostic messages.
...	options for LFIread.

**Details**

As a user, you will probably only need to use LFIopen and LFIdec. The other functions are called implicitly. LFI files don't store all date information. Only the valid date is available (forecast date + range). So there is no way to know the forecast date and the lead time separately, except e.g. from the file name, but that is up to the user.

**Value**

An LFIfile object is a list with two components:

- listA list of all available fields: name and (byte) loxcation in the file. The columns contain byte address etc.
- holesNot implemented for LFI. NULL.

A LFIfile object also has a list of attributes:



- **header**The LFI header: a vector of 22 integers.
- **filename**Full name of the file.
- **nfields**Number of fields contained in the file.
- **domain**All geographical meta data: domain size, projection, vertical levels etc.
- **validdate**Time information. LFI files contain no information about leadtime!

LFIdec returns a `geofield`, but note that the date information is minimal, because LFI files don't even store forecast lead time separately.

### See Also

[LFIreplace](#), [FAopen](#), [FAdec](#)

---

LFIreplace	<i>Encode an LFI field (SURFEX). Only replacement of an existing field is possible.</i>
------------	---

---

### Description

Encodes a matrix to the given field in a LFI file.

### Usage

```
LFIreplace(lfi,field,data,missing=1.0E+20,quiet=TRUE)
```

### Arguments

<b>lfi</b>	An object as returned by <code>LFIopen</code> . <code>lfi</code> may also be a character string, in which case it is interpreted as a filename (and automatically opened using <code>LFIopen</code> ).
<b>field</b>	Can be either an integer, in which case it indicates the position of the field in the fieldlist, or else it can be a character string. If <code>field</code> is a string, it may any part of the complete name of the field, as long as it is unique.
<b>data</b>	A matrix with the new values. It must have the right dimensions of the original field. LFI fields have a "skin" of missing values that surround the data. Thus, <code>data</code> must have either dimensions <code>(nx,ny)</code> , in which case NA values are added at the border, or <code>(nx+2,ny+2)</code> .
<b>missing</b>	Data value that should be interpreted as missing value. NA values in <code>data</code> are replaced by this value before encoding.
<b>quiet</b>	FALSE turns on the diagnostic messages.

### Details

The field values are replaced in the LFI file by the given data.

**Value**

The return value is a `LFIfile` class object with the new contents.

**See Also**

[LFiopen](#), [LFidec](#)

---

utilities

*Simple utilities*

---

**Description**

Various functions that may make life easier.

**Usage**

```
FAdescribe(fname)
FAsizes(nmin,nmax)
FAfind(fa,field)
```

**Arguments**

<code>fname</code>	The name of a field in a FA file. A string of length 16
<code>nmin,nmax</code>	Minimum and maximum of the vector of admissible FA grid sizes.
<code>fa</code>	A FAfile
<code>field</code>	An integer or the name of a field.

**Value**

- `FAsizes` returns a vector of admissible FA grid sizes (powers of 2,3,5).
- `FAdescribe` returns a 'description' by splitting a FA field name into level information and parameter name.
- `FAfind` searches for a field name in the list of fields in a FAfile. This uses `grep`, so giving less than 16 characters may work, but it may return multiple matches.

# Index

## \*Topic **file**

- FAcreate, [3](#)
  - FAdec, [4](#)
  - FAechkevo, [5](#)
  - FAenc, [6](#)
  - FAframe, [8](#)
  - FAmake.frame, [9](#)
  - FAmake.header, [10](#)
  - FAmake.time, [10](#)
  - FAopen, [11](#)
  - FApressure, [12](#)
  - FAslice, [13](#)
  - FAsounding, [14](#)
  - internal, [15](#)
  - LFIopen, [16](#)
  - LFIreplace, [17](#)
  - utilities, [18](#)
- biper, [2](#), [7](#)
- ecto, [2](#)
- FAadd (FAenc), [6](#)
  - FAcreate, [3](#), [9–11](#)
  - FAdec, [4](#), [6](#), [7](#), [12](#), [17](#)
  - FAdescribe (utilities), [18](#)
  - FAdomain (FAframe), [8](#)
  - FAechk.open (FAechkevo), [5](#)
  - FAechk.read (FAechkevo), [5](#)
  - FAechkevo, [5](#)
  - FAenc, [5](#), [6](#)
  - FAfft2raw (internal), [15](#)
  - FAfind (utilities), [18](#)
  - FAframe, [8](#)
  - FAheader.date (internal), [15](#)
  - FAheader.rewrite (internal), [15](#)
  - FAmake.frame, [3](#), [9](#), [10](#), [11](#)
  - FAmake.header, [9](#), [10](#), [11](#)
  - FAmake.time, [9](#), [10](#), [10](#)
  - FAMsg.dec (FAdec), [4](#)
  - FAMsg.enc (FAenc), [6](#)
  - FAopen, [3](#), [5](#), [7](#), [8](#), [11](#), [12](#), [17](#)
  - FAopenTar (FAopen), [11](#)
  - FApressure, [12](#), [13](#), [14](#)
  - FApressures.local (FApressure), [12](#)
  - FAraw2fft (internal), [15](#)
  - FArawreorder (internal), [15](#)
  - FAread.meta (FAframe), [8](#)
  - FAremove (FAenc), [6](#)
  - FAsizes (utilities), [18](#)
  - FAslice, [13](#)
  - FAsounding, [13](#), [14](#)
  - FAtime (FAframe), [8](#)
- internal, [15](#)
- iview, [5](#), [12](#)
- LFIdec, [18](#)
  - LFIdec (LFIopen), [16](#)
  - LFIdomain (LFIopen), [16](#)
  - LFIopen, [16](#), [18](#)
  - LFIread (LFIopen), [16](#)
  - LFIreplace, [17](#), [17](#)
  - LFItime (LFIopen), [16](#)
- point.interp, [14](#)
- utilities, [18](#)