

MI - Fimex

Generated by Doxygen 1.5.5

Fri May 7 15:50:08 2010

Contents

1	Fimex User Documentation	1
1.1	Setup Files	1
2	fimex Program Options	3
2.1	fimex Program Options	4
3	Configuration files for felt reader	7
4	ncml Configuration	9
4.1	ncml Configuration	10
5	quality-extraction Configuration	11
5.1	quality-extraction Configuration	12
6	gribWriter Configuration	13
6.1	gribWriter Configuration	14
7	netcdfWriter Configuration	15
7.1	netcdfWriter Configuration	16
8	Deprecated List	17
9	Namespace Index	19
9.1	Namespace List	19
10	Class Index	21
10.1	Class Hierarchy	21
11	Class Index	23
11.1	Class List	23
12	File Index	25
12.1	File List	25

13 Namespace Documentation	27
13.1 felt Namespace Reference	27
13.2 MetNoFelt Namespace Reference	30
13.3 MetNoFimex Namespace Reference	32
14 Class Documentation	43
14.1 binary< N > Struct Template Reference	43
14.2 binary< 0 > Struct Template Reference	44
14.3 MetNoFimex::C_CDMReader Class Reference	45
14.4 MetNoFimex::CachedForwardInterpolation Class Reference	47
14.5 MetNoFimex::CachedInterpolation Class Reference	48
14.6 MetNoFimex::CachedInterpolationInterface Class Reference	50
14.7 MetNoFimex::CachedVectorReprojection Class Reference	51
14.8 MetNoFimex::CDM Class Reference	52
14.9 MetNoFimex::CDMAttribute Class Reference	65
14.10MetNoFimex::CDMDimension Class Reference	69
14.11MetNoFimex::CDMException Class Reference	71
14.12MetNoFimex::CDMExtractor Class Reference	72
14.13MetNoFimex::CDMFileReaderFactory Class Reference	75
14.14MetNoFimex::CDMInterpolator Class Reference	77
14.15MetNoFimex::CDMNameCompare Struct Reference	80
14.16MetNoFimex::CDMNamedEntity Class Reference	81
14.17MetNoFimex::CDMNameEqual Class Reference	82
14.18MetNoFimex::CDMNameEqualPtr Class Reference	83
14.19MetNoFimex::CDMQualityExtractor Class Reference	84
14.20MetNoFimex::CDMReader Class Reference	86
14.21MetNoFimex::CDMTimeInterpolator Class Reference	92
14.22MetNoFimex::CDMVariable Class Reference	94
14.23MetNoFimex::CDMWriter Class Reference	97
14.24MetNoFimex::ChangeMissingValue< IN, OUT > Class Template Reference	98
14.25MetNoFimex::CompleteCoordinateSystemForComparator Struct Reference	99
14.26MetNoFimex::CoordinateAxis Class Reference	100
14.27MetNoFimex::CoordinateSystem Class Reference	102
14.28MetNoFimex::Data Class Reference	107
14.29MetNoFimex::DataImpl< C > Class Template Reference	112
14.30MetNoFimex::DataTypeChanger Class Reference	120
14.31MetNoFelt::Felt_Array Class Reference	122

14.32MetNoFelt::Felt_Array2 Class Reference	127
14.33MetNoFelt::Felt_File Class Reference	131
14.34MetNoFelt::Felt_File2 Class Reference	135
14.35MetNoFelt::Felt_File_Error Class Reference	139
14.36MetNoFimex::FeltCDMReader Class Reference	140
14.37MetNoFimex::FeltCDMReader2 Class Reference	141
14.38felt::FeltField Class Reference	142
14.39felt::FeltFile Class Reference	145
14.40felt::FeltGridDefinition Class Reference	148
14.41MetNoFelt::FeltParameters Class Reference	150
14.42MetNoFimex::FimexTime Class Reference	152
14.43MetNoFimex::GribApiCDMWriter Class Reference	155
14.44MetNoFimex::GribApiCDMWriter_Impl1 Class Reference	156
14.45MetNoFimex::GribApiCDMWriter_Impl2 Class Reference	158
14.46MetNoFimex::GribApiCDMWriter_ImplAbstract Class Reference	160
14.47MetNoFimex::GribCDMReader Class Reference	164
14.48MetNoFimex::GribFileIndex Class Reference	165
14.49MetNoFimex::GribFileMessage Class Reference	166
14.50MetNoFimex::GribFileMessageEqualLevelTime Class Reference	169
14.51MetNoFimex::GribFileMessageEqualTime Class Reference	170
14.52MetNoFimex::GridDefinition Class Reference	171
14.53MetNoFimex::LambertConformalConicProjection Class Reference	175
14.54MetNoFimex::LatitudeLongitudeProjection Class Reference	177
14.55MetNoFelt::LevelPairLess Struct Reference	179
14.56MetNoFimex::Logger Class Reference	180
14.57MetNoFimex::MercatorProjection Class Reference	182
14.58mifi_cdm_reader Class Reference	184
14.59MetNoFimex::NcmlCDMReader Class Reference	185
14.60MetNoFimex::NetCDF_CDMReader Class Reference	187
14.61MetNoFimex::NetCDF_CDMWriter Class Reference	189
14.62MetNoFimex::Null_CDMWriter Class Reference	191
14.63MetNoFimex::PolarStereographicProjection Class Reference	192
14.64MetNoFimex::Projection Class Reference	194
14.65MetNoFimex::ProjectionImpl Class Reference	197
14.66MetNoFimex::ReplaceStringObject Class Reference	201
14.67MetNoFimex::ReplaceStringTimeObject Class Reference	203

14.68MetNoFimex::RotatedLatitudeLongitudeProjection Class Reference	205
14.69MetNoFimex::ScaleValue< IN, OUT > Class Template Reference	207
14.70MetNoFelt::ShortPairLess Struct Reference	208
14.71MetNoFimex::SliceBuilder Class Reference	209
14.72MetNoFimex::SpatialAxisSpec Class Reference	211
14.73MetNoFimex::staticCast< OUT > Struct Template Reference	213
14.74MetNoFimex::StereographicProjection Class Reference	214
14.75MetNoFimex::TimeLevelDataSliceFetcher Class Reference	216
14.76MetNoFimex::TimeSpec Class Reference	217
14.77MetNoFimex::TimeUnit Class Reference	219
14.78MetNoFimex::UnitException Class Reference	221
14.79MetNoFimex::Units Class Reference	222
14.80MetNoFimex::XMLDoc Class Reference	224
15 File Documentation	227
15.1 doxydoc.txt File Reference	227
15.2 include/felt/FeltConstants.h File Reference	228
15.3 include/felt/FeltField.h File Reference	229
15.4 include/felt/FeltFile.h File Reference	230
15.5 include/felt/FeltGridDefinition.h File Reference	231
15.6 include/felt/FeltTypeConversion.h File Reference	232
15.7 include/felt/FeltTypes.h File Reference	233
15.8 include/fimex/binaryConstants.h File Reference	234
15.9 include/fimex/C_CDMReader.h File Reference	235
15.10include/fimex/c_fimex.h File Reference	236
15.11include/fimex/CachedForwardInterpolation.h File Reference	242
15.12include/fimex/CachedInterpolation.h File Reference	243
15.13include/fimex/CachedVectorReprojection.h File Reference	244
15.14include/fimex/CDM.h File Reference	245
15.15include/fimex/CDMAttribute.h File Reference	246
15.16include/fimex/CDMconstants.h File Reference	247
15.17include/fimex/CDMDataType.h File Reference	249
15.18include/fimex/CDMDimension.h File Reference	250
15.19include/fimex/CDMException.h File Reference	251
15.20include/fimex/CDMExtractor.h File Reference	252
15.21include/fimex/CDMFileReaderFactory.h File Reference	253
15.22include/fimex/CDMInterpolator.h File Reference	254

15.23include/fimex/CDMNamedEntity.h File Reference	255
15.24include/fimex/CDMQualityExtractor.h File Reference	256
15.25include/fimex/CDMReader.h File Reference	257
15.26include/fimex/CDMTimeInterpolator.h File Reference	258
15.27include/fimex/CDMVariable.h File Reference	259
15.28include/fimex/CDMWriter.h File Reference	260
15.29include/fimex/coordSys/CoordinateAxis.h File Reference	261
15.30include/fimex/coordSys/CoordinateSystem.h File Reference	262
15.31include/fimex/coordSys/LambertConformalConicProjection.h File Reference	263
15.32include/fimex/coordSys/LatitudeLongitudeProjection.h File Reference	264
15.33include/fimex/coordSys/MercatorProjection.h File Reference	265
15.34include/fimex/coordSys/PolarStereographicProjection.h File Reference	266
15.35include/fimex/coordSys/Projection.h File Reference	267
15.36include/fimex/coordSys/ProjectionImpl.h File Reference	268
15.37include/fimex/coordSys/RotatedLatitudeLongitudeProjection.h File Reference	269
15.38include/fimex/coordSys/StereographicProjection.h File Reference	270
15.39include/fimex/Data.h File Reference	271
15.40include/fimex/DataImpl.h File Reference	272
15.41include/fimex/DataTypeChanger.h File Reference	274
15.42include/fimex/deprecated.h File Reference	275
15.43include/fimex/Felt_Array.h File Reference	276
15.44include/fimex/Felt_Array2.h File Reference	277
15.45include/fimex/Felt_File.h File Reference	278
15.46include/fimex/Felt_File2.h File Reference	279
15.47include/fimex/Felt_File_Error.h File Reference	280
15.48include/fimex/Felt_Types.h File Reference	281
15.49include/fimex/FeltCDMReader.h File Reference	282
15.50include/fimex/FeltCDMReader2.h File Reference	283
15.51include/fimex/FeltParameters.h File Reference	284
15.52include/fimex/GribApiCDMWriter.h File Reference	285
15.53include/fimex/GribApiCDMWriter_Impl1.h File Reference	286
15.54include/fimex/GribApiCDMWriter_Impl2.h File Reference	287
15.55include/fimex/GribApiCDMWriter_ImplAbstract.h File Reference	288
15.56include/fimex/GribCDMReader.h File Reference	289
15.57include/fimex/GribFileIndex.h File Reference	290
15.58include/fimex/GribUtils.h File Reference	291

15.59include/fimex/GridDefinition.h File Reference	292
15.60include/fimex/interpolation.h File Reference	293
15.61include/fimex/Logger.h File Reference	304
15.62include/fimex/mifi_cdm_reader.h File Reference	305
15.63include/fimex/NcmlCDMReader.h File Reference	306
15.64include/fimex/NetCDF_CDMReader.h File Reference	307
15.65include/fimex/NetCDF_CDMWriter.h File Reference	308
15.66include/fimex/Null_CDMWriter.h File Reference	309
15.67include/fimex/ReplaceStringObject.h File Reference	310
15.68include/fimex/ReplaceStringTimeObject.h File Reference	311
15.69include/fimex/SliceBuilder.h File Reference	312
15.70include/fimex/SpatialAxisSpec.h File Reference	313
15.71include/fimex/TimeLevelDataSliceFetcher.h File Reference	314
15.72include/fimex/TimeSpec.h File Reference	315
15.73include/fimex/TimeUnit.h File Reference	316
15.74include/fimex/Units.h File Reference	317
15.75include/fimex/Utils.h File Reference	318
15.76include/fimex/XMLDoc.h File Reference	319
16 Example Documentation	321
16.1 coordinateSystem.cpp	321

Chapter 1

Fimex User Documentation

Fimex is a the File Interpolation, Manipulation and EXtraction library for gridded geospatial data. It converts between different, extensible dataformats (currently netcdf, grib1/2 and [felt](#)). It enables you to change the projection and interpolation of scalar and vector grids. It makes it possible subset the gridded data and to extract only parts of the files.

Fimex can be used as library called *Fimex* and a command-line program called *fimex*, which gives access to most but not all functions of the library.

Fimex is build around the Common Data Model version 1 developed by Unidata and uses a describes data using the CF-Convention <http://cf-pcmdi.llnl.gov/documents/cf-conventions/1.0/cf-conventions.html>. Knowledge of that convention is not required, but will help understanding the config files needed for conversion.

The API of Fimex as included in this document is not stable yet and can change without warning. The setup-files are considered to be mostly stable. The fimex-program can thus savely be used. If you want to use the API, please contact me.

1.1 Setup Files

Detailed information on the differnt configuration files can be found at:

- [fimex Program Options](#)
- [Configuration files for felt reader](#)
- [ncml Configuration](#)
- [gribWriter Configuration](#)
- [netcdfWriter Configuration](#)
- [quality-extraction Configuration](#)

Chapter 2

fimex Program Options

2.1 fimex Program Options

fimex is a command-line program. It has the following options:

```
usage: fimex --input.file  FILENAME [--input.type  INPUT_TYPE]
        --output.file  FILENAME [--output.type  OUTPUT_TYPE]
        [--input.config  CFGFILENAME] [--output.config  CFGFILENAME]
        [--extract....]
        [--interpolate....]
        [--timeInterpolate....]
```

Generic options:

-h [--help]	help message
--version	program version
--debug	debug program
--print-options	print all options
-c [--config] arg (=fimex.cfg)	configuration file

Configurational options:

--input.file arg	input file
--input.type arg	filetype of input file
--input.config arg	non-standard input configuration
--input.printNcML	print NcML description of input file
--input.printCS	print CoordinateSystem
--output.file arg	output file
--output.type arg	filetype of output file
--output.config arg	non-standard output configuration
--qualityExtract.autoConfString arg	quality-restrictions using CF-1.3 and a flag
--qualityExtract.config arg	quality-restrictions configuration file
--qualityExtract.printNcML	print NcML description of qualityExtractor
--quality.printCS	print CoordinateSystem
--extract.removeVariable arg	remove variables
--extract.reduceDimension.name arg	name of a dimension to reduce
--extract.reduceDimension.start arg	start position of the dimension to reduce (>=0)
--extract.reduceDimension.end arg	end position of the dimension to reduce
--extract.printNcML	print NcML description of extractor
--extract.printCS	print CoordinateSystem
--interpolate.projString arg	proj4 input string describing the new projection
--interpolate.method arg	interpolation method, one of nearestneighbor, bilinear or bicubic
--interpolate.xAxisValues arg	string with values on x-Axis, use ... to continue, i.e. 10.5,11,...,29.5 see Fimex::SpatialAxisSpec for full definition
--interpolate.yAxisValues arg	string with values on y-Axis, use ... to continue, i.e. 10.5,11,...,29.5 see Fimex::SpatialAxisSpec for full definition
--interpolate.xAxisUnit arg	unit of x-Axis given as udunits string, i.e. m or degrees_east
--interpolate.yAxisUnit arg	unit of y-Axis given as udunits string, i.e. m or degrees_north
--interpolate.latitudeName arg	name for auto-generated projection coordinate latitude
--interpolate.longitudeName arg	name for auto-generated projection coordinate longitude
--interpolate.printNcML	print NcML description of interpolator
--interpolate.printCS	print CoordinateSystem
--timeInterpolate.timeSpec arg	specification of times to interpolate to, see Fimex::TimeSpec for a full definition
--timeInterpolate.printNcML	print NcML description of timeInterpolator
--timeInterpolate.printCS	print CoordinateSystem
--ncml.config	modify/configure with ncml-file
--ncml.printNcML	print NcML description after ncml-configuration
--ncml.printCS	print CoordinateSystem

All the configurational options can be configured using a configuration file which is supplied using the `--config` option. All command line options (CLO) will overwrite the config-file. As a rule of thumb, use the CLO for testing and use the config-file for productive usage. The CLOs will be further explained in [fimex](#)

[Setup File](#).

2.1.1 fimex Setup File

The *SpatialAxisSpec* used in *xAxisValues* or *yAxisValues* for the spatial interpolation should be formatted as explained in detail in [MetNoFimex::SpatialAxisSpec](#). It allows also autotuning to the original data-values.

The *TimeSpec* string used for the *timeInterpolate* should be formatted as explained in detail in [Met-NoFimex::TimeSpec](#).

Chapter 3

Configuration files for felt reader

The xml configuration files are defined by the *felt2nc_variables.dtd* definition. Since part of this configuration are quite stable, e.g. the axes (time, level, lat, lon, x, y), other parts change, e.g. the variables to translate change very often. It is therefore useful to split the variables from the rest of the configuration via *xinclude*

When writing a new configuration for a new set of felt-files, usually from a new model, it is wise to group the configuration by

1. time resolution, i.e. one config for 3hourly files, one config for hourly files
2. spatial resolution: *fimex* doesn't allow different spatial resolutions, but some models use coarser resolution for higher levels
3. vertical levels: it is difficult to have the same parameter with sigma levels and with height in m

Grouping can be done in two ways, the first one being faster in operation, the second is easier to configure/change consistently:

1. write different configuration-files for each group of parameter, stating the parameter as well as possible.
2. write one configuration-file for all parameter, keeping the parameters as variable as possible. Use a preprocess-step to extract each group. Use e.g. *nyfelt* or *felt2felt* as preprocessor

By default, all data is read as *type="short"* data with a scaling factor. While *felt* allows for one scaling factor for each timestep, height and parameter, the CDM allows only for one scaling factor per parameter. When the scaling factor changes withing height or timestep, *fimex* will fail to read the data as short. It is therefore useful to read data as *type="float"*, which will automatically expand the scaling factor. If the resulting file is to big, it is possible to convert to short with one scaling factor and offset using the [netcdfWriter Configuration](#).

Before running *fimex* with a new *felt* configuration, make sure the file is valid, e.g. with

```
xmllint --valid --noout felt2nc_config.xml
```

Unfortunately, *xinclude* and validation don't play well together, since usual validation happens before the inclusion of external parts. *xmllint* uses special options to fix those problem:

```
xmllint --xinclude --postvalid --noout felt2nc_config.xml
```

Below follows a complete felt-configuration.

Chapter 4

ncml Configuration

4.1 ncml Configuration

Unidata's NetCDF Markup Language (NcML) as described in <http://www.unidata.ucar.edu/software/netcdf/ncml/> gives the opportunity to change all information written in the CDM. With the `-ncml.config` option, the CDM will be configured immediately after reading a file. It is also possible to read in a ncml file with the `-input.file=xxx.ncml` option. In this case, the real data must be linked with the 'location' markup.

Input-files can and should be validated against the included ncml-2.2.xsd.

Warning:

not all features are supported in the current implementation. Missing features are: aggregation, changing dimension sizes, unlimited dimensions, adding/changing values of variables, groups

Chapter 5

quality-extraction Configuration

5.1 quality-extraction Configuration

Warning:

The quality-extraction is still in a very early stage of development. The configuration and the outcome is very likely to change in further developments. Any feedback is strongly welcome.

In cases where the data should be extracted if certain conditions (qualities) apply, i.e. the status-flag indicates a properly working instrument, or the sea-surface-temperature is above 300K, the [Met-NoFimex::CDMQualityExtractor](#) allows to add these rules. The `cdmQualityConfig.xml` file as shown above gives an example of such an configuration.

- The variable "bla" will only be set, if "blub" has integer-values between 1 and 6.
- The variable "air_temperature" will only be extracted for an "altitude" above 1000. The value 1000 is the actual data value in the variable "altitude" without any scaling or unit-conversion applied.

The following use-values can be selected:

- `all` select all valid values (within `valid_max`, `valid_min` or `valid_range`, without `_FillValue`)
- `highest` the highest numerical value found in the data-slice which is valid
- `lowest` the lowest numerical value found in the data-slice which is valid
- `max:xxx.x` all valid-values below or equal `xxx.x`
- `min:xxx.x` all valid values above or equal `xxx.x`

All values which do not match the quality-criteria will be set to the `_FillValue` of the variable.

Chapter 6

gribWriter Configuration

6.1 gribWriter Configuration

Chapter 7

netcdfWriter Configuration

7.1 netcdfWriter Configuration

The netcdfWriterConfig gives the opportunity to set some features explicit only for netcdf-files, i.e. file-format (netcdf3/4) or compression.

It is also possible to add an [ncml Configuration](#) to the output to change the internal structure just before writing.

It is also possible to change units including all value in the netcdfWriterDoc. Changing the units in the ncmlConfiguration would change the attribute value only, but not the data.

The CDM resembles a netcdf datastructure. In general, there is no need to use a configuration for this writer, but it might be useful in the following cases:

- Output-files are too big, and a change of datatype i.e. from float to short is desired
- Different attributes are required for special usages, but the input-configuration of the reader shouldn't be changed.
- Different variable or dimension names are required for special usages.

Chapter 8

Deprecated List

Member **MetNoFimex::CDM::DEPRECATED**(AttrVec getProjection(std::string varName) const)
use the getProjectionOf() method

Member **MetNoFimex::DEPRECATED**(std::string attributesToProjString(const std::vector< CDMAtribute > &attrs)
use Projection::create() with Projection::getProj4String() instead

Member **MetNoFimex::DEPRECATED**(std::vector< CDMAtribute > projStringToAttributes(std::string projStr))
use Projection::createByProj4() and Projection::getParameters()

Chapter 9

Namespace Index

9.1 Namespace List

Here is a list of all namespaces with brief descriptions:

felt	27
MetNoFelt	30
MetNoFimex	32

Chapter 10

Class Index

10.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

binary< N >	43
binary< 0 >	44
std::binary_function< _Arg1, _Arg2, _Result >[external]	
MetNoFelt::LevelPairLess	179
MetNoFelt::ShortPairLess	208
MetNoFimex::CDMNameCompare	80
MetNoFimex::CachedInterpolationInterface	50
MetNoFimex::CachedForwardInterpolation	47
MetNoFimex::CachedInterpolation	48
MetNoFimex::CachedVectorReprojection	51
MetNoFimex::CDM	52
MetNoFimex::CDMFileReaderFactory	75
MetNoFimex::CDMNamedEntity	81
MetNoFimex::CDMAttribute	65
MetNoFimex::CDMDimension	69
MetNoFimex::CDMVariable	94
MetNoFimex::CoordinateAxis	100
MetNoFimex::Projection	194
MetNoFimex::ProjectionImpl	197
MetNoFimex::LambertConformalConicProjection	175
MetNoFimex::LatitudeLongitudeProjection	177
MetNoFimex::MercatorProjection	182
MetNoFimex::RotatedLatitudeLongitudeProjection	205
MetNoFimex::StereographicProjection	214
MetNoFimex::PolarStereographicProjection	192
MetNoFimex::CDMReader	86
MetNoFimex::C_CDMReader	45
MetNoFimex::CDMExtractor	72
MetNoFimex::CDMInterpolator	77
MetNoFimex::CDMQualityExtractor	84
MetNoFimex::CDMTimeInterpolator	92
MetNoFimex::FeltCDMReader	140

MetNoFimex::FeltCDMReader2	141
MetNoFimex::GribCDMReader	164
MetNoFimex::NcmlCDMReader	185
MetNoFimex::NetCDF_CDMReader	187
MetNoFimex::CDMWriter	97
MetNoFimex::GribApiCDMWriter	155
MetNoFimex::GribApiCDMWriter_ImplAbstract	160
MetNoFimex::GribApiCDMWriter_Impl1	156
MetNoFimex::GribApiCDMWriter_Impl2	158
MetNoFimex::NetCDF_CDMWriter	189
MetNoFimex::Null_CDMWriter	191
MetNoFimex::CoordinateSystem	102
MetNoFimex::Data	107
MetNoFimex::DataImpl< C >	112
MetNoFimex::DataTypeChanger	120
std::exception[external]	
std::runtime_error[external]	
MetNoFelt::Felt_File_Error	139
MetNoFimex::CDMException	71
MetNoFimex::UnitException	221
MetNoFelt::Felt_Array	122
MetNoFelt::Felt_Array2	127
MetNoFelt::Felt_File	131
MetNoFelt::Felt_File2	135
felt::FeltField	142
felt::FeltFile	145
felt::FeltGridDefinition	148
MetNoFelt::FeltParameters	150
MetNoFimex::FimexTime	152
MetNoFimex::GribFileIndex	165
MetNoFimex::GribFileMessage	166
MetNoFimex::GridDefinition	171
MetNoFimex::Logger	180
mifi_cdm_reader	184
MetNoFimex::ReplaceStringObject	201
MetNoFimex::ReplaceStringTimeObject	203
MetNoFimex::SliceBuilder	209
MetNoFimex::SpatialAxisSpec	211
MetNoFimex::staticCast< OUT >	213
MetNoFimex::TimeLevelDataSliceFetcher	216
MetNoFimex::TimeSpec	217
MetNoFimex::TimeUnit	219
std::unary_function< _Arg, _Result >[external]	
MetNoFimex::CDMNameEqual	82
MetNoFimex::CDMNameEqualPtr	83
MetNoFimex::CompleteCoordinateSystemForComparator	99
MetNoFimex::GribFileMessageEqualLevelTime	169
MetNoFimex::GribFileMessageEqualTime	170
std::unary_function< IN, OUT >[external]	
MetNoFimex::ChangeMissingValue< IN, OUT >	98
MetNoFimex::ScaleValue< IN, OUT >	207
MetNoFimex::Units	222
MetNoFimex::XMLDoc	224

Chapter 11

Class Index

11.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

binary< N >	43
binary< 0 >	44
MetNoFimex::C_CDMReader	45
MetNoFimex::CachedForwardInterpolation	47
MetNoFimex::CachedInterpolation	48
MetNoFimex::CachedInterpolationInterface	50
MetNoFimex::CachedVectorReprojection	51
MetNoFimex::CDM (Data structure of the Common Data Model)	52
MetNoFimex::CDMAttribute	65
MetNoFimex::CDMDimension	69
MetNoFimex::CDMException	71
MetNoFimex::CDMExtractor	72
MetNoFimex::CDMFileReaderFactory	75
MetNoFimex::CDMInterpolator	77
MetNoFimex::CDMNameCompare	80
MetNoFimex::CDMNamedEntity	81
MetNoFimex::CDMNameEqual	82
MetNoFimex::CDMNameEqualPtr	83
MetNoFimex::CDMQualityExtractor (Extract data with defined quality status)	84
MetNoFimex::CDMReader (Basic interface for CDM reading and manipulation classes)	86
MetNoFimex::CDMTimeInterpolator	92
MetNoFimex::CDMVariable	94
MetNoFimex::CDMWriter	97
MetNoFimex::ChangeMissingValue< IN, OUT >	98
MetNoFimex::CompleteCoordinateSystemForComparator	99
MetNoFimex::CoordinateAxis	100
MetNoFimex::CoordinateSystem	102
MetNoFimex::Data	107
MetNoFimex::DataImpl< C >	112
MetNoFimex::DataTypeChanger	120
MetNoFelt::Felt_Array (Encapsulate parameters of a felt file)	122
MetNoFelt::Felt_Array2 (A collection of FeltFields to build a 4-dimensional data array for 1-parameter and 1-vertical coordinate)	127

MetNoFelt::Felt_File (Felt File access)	131
MetNoFelt::Felt_File2 (Felt File access)	135
MetNoFelt::Felt_File_Error	139
MetNoFimex::FeltCDMReader	140
MetNoFimex::FeltCDMReader2	141
felt::FeltField	142
felt::FeltFile	145
felt::FeltGridDefinition	148
MetNoFelt::FeltParameters	150
MetNoFimex::FimexTime	152
MetNoFimex::GribApiCDMWriter	155
MetNoFimex::GribApiCDMWriter_Impl1	156
MetNoFimex::GribApiCDMWriter_Impl2	158
MetNoFimex::GribApiCDMWriter_ImplAbstract	160
MetNoFimex::GribCDMReader	164
MetNoFimex::GribFileIndex	165
MetNoFimex::GribFileMessage	166
MetNoFimex::GribFileMessageEqualLevelTime (Functor to find messages with equal level and time)	169
MetNoFimex::GribFileMessageEqualTime (Functor to find Messages with equal time)	170
MetNoFimex::GridDefinition	171
MetNoFimex::LambertConformalConicProjection	175
MetNoFimex::LatitudeLongitudeProjection	177
MetNoFelt::LevelPairLess	179
MetNoFimex::Logger	180
MetNoFimex::MercatorProjection	182
mifi_cdm_reader	184
MetNoFimex::NcmlCDMReader	185
MetNoFimex::NetCDF_CDMReader	187
MetNoFimex::NetCDF_CDMWriter	189
MetNoFimex::Null_CDMWriter	191
MetNoFimex::PolarStereographicProjection	192
MetNoFimex::Projection	194
MetNoFimex::ProjectionImpl	197
MetNoFimex::ReplaceStringObject	201
MetNoFimex::ReplaceStringTimeObject	203
MetNoFimex::RotatedLatitudeLongitudeProjection	205
MetNoFimex::ScaleValue< IN, OUT >	207
MetNoFelt::ShortPairLess	208
MetNoFimex::SliceBuilder	209
MetNoFimex::SpatialAxisSpec	211
MetNoFimex::staticCast< OUT >	213
MetNoFimex::StereographicProjection	214
MetNoFimex::TimeLevelDataSliceFetcher (Read a slice of a given time/level combination from a cdmReader)	216
MetNoFimex::TimeSpec	217
MetNoFimex::TimeUnit	219
MetNoFimex::UnitException	221
MetNoFimex::Units	222
MetNoFimex::XMLDoc	224

Chapter 12

File Index

12.1 File List

Here is a list of all files with brief descriptions:

include/felt/FeltConstants.h	228
include/felt/FeltField.h	229
include/felt/FeltFile.h	230
include/felt/FeltGridDefinition.h	231
include/felt/FeltTypeConversion.h	232
include/felt/FeltTypes.h	233
include/fimex/binaryConstants.h	234
include/fimex/C_CDMReader.h	235
include/fimex/c_fimex.h	236
include/fimex/CachedForwardInterpolation.h	242
include/fimex/CachedInterpolation.h	243
include/fimex/CachedVectorReprojection.h	244
include/fimex/CDM.h	245
include/fimex/CDMAttribute.h	246
include/fimex/CDMconstants.h	247
include/fimex/CDMDataType.h	249
include/fimex/CDMDimension.h	250
include/fimex/CDMException.h	251
include/fimex/CDMExtractor.h	252
include/fimex/CDMFileReaderFactory.h	253
include/fimex/CDMInterpolator.h	254
include/fimex/CDMNamedEntity.h	255
include/fimex/CDMQualityExtractor.h	256
include/fimex/CDMReader.h	257
include/fimex/CDMTimeInterpolator.h	258
include/fimex/CDMVariable.h	259
include/fimex/CDMWriter.h	260
include/fimex/Data.h	271
include/fimex/DataImpl.h	272
include/fimex/DataTypeChanger.h	274
include/fimex/deprecated.h	275
include/fimex/Felt_Array.h	276
include/fimex/Felt_Array2.h	277

include/fimex/Felt_File.h	278
include/fimex/Felt_File2.h	279
include/fimex/Felt_File_Error.h	280
include/fimex/Felt_Types.h	281
include/fimex/FeltCDMReader.h	282
include/fimex/FeltCDMReader2.h	283
include/fimex/FeltParameters.h	284
include/fimex/GribApiCDMWriter.h	285
include/fimex/GribApiCDMWriter_Impl1.h	286
include/fimex/GribApiCDMWriter_Impl2.h	287
include/fimex/GribApiCDMWriter_ImplAbstract.h	288
include/fimex/GribCDMReader.h	289
include/fimex/GribFileIndex.h	290
include/fimex/GribUtils.h	291
include/fimex/GridDefinition.h	292
include/fimex/interpolation.h	293
include/fimex/Logger.h	304
include/fimex/mifi_cdm_reader.h	305
include/fimex/NcmlCDMReader.h	306
include/fimex/NetCDF_CDMReader.h	307
include/fimex/NetCDF_CDMWriter.h	308
include/fimex/Null_CDMWriter.h	309
include/fimex/ReplaceStringObject.h	310
include/fimex/ReplaceStringTimeObject.h	311
include/fimex/SliceBuilder.h	312
include/fimex/SpatialAxisSpec.h	313
include/fimex/TimeLevelDataSliceFetcher.h	314
include/fimex/TimeSpec.h	315
include/fimex/TimeUnit.h	316
include/fimex/Units.h	317
include/fimex/Utils.h	318
include/fimex/XMLDoc.h	319
include/fimex/coordSys/CoordinateAxis.h	261
include/fimex/coordSys/CoordinateSystem.h	262
include/fimex/coordSys/LambertConformalConicProjection.h	263
include/fimex/coordSys/LatitudeLongitudeProjection.h	264
include/fimex/coordSys/MercatorProjection.h	265
include/fimex/coordSys/PolarStereographicProjection.h	266
include/fimex/coordSys/Projection.h	267
include/fimex/coordSys/ProjectionImpl.h	268
include/fimex/coordSys/RotatedLatitudeLongitudeProjection.h	269
include/fimex/coordSys/StereographicProjection.h	270

Chapter 13

Namespace Documentation

13.1 felt Namespace Reference

Classes

- class [FeltField](#)
- class [FeltFile](#)
- class [FeltGridDefinition](#)

Typedefs

- typedef short int [word](#)
A [felt](#) block "word" - 2 bytes.
- typedef boost::shared_ptr< [FeltGridDefinition](#) > [FeltGridDefinitionPtr](#)

Functions

- bool [isUndefined](#) ([word](#) w)
- boost::array< float, 6 > [gridParameters](#) (int gridType, int xNum, int yNum, int a, int b, int c, int d, const **std::vector**< short int > &extraData)
- **std::string** [gridParametersToProjDefinition](#) (int gridType, const boost::array< float, 6 > &gridPars)
- **std::ostream** & [contentSummary](#) (**std::ostream** &out, const [FeltGridDefinition](#) &grid)
- template<typename T>
T [get](#) ([word](#) w)
- boost::posix_time::ptime [parseTime](#) (const [word](#) *data)
- boost::posix_time::ptime [parseTimeNoThrow](#) (const [word](#) *data)

Variables

- const size_t [blockWords](#) = 1024
- const size_t [blockSize](#) = [blockWords](#) * sizeof([word](#))
- const size_t [offsetToContentDefinition](#) = 2
- const double [PI](#) = M_PI
- const double [EARTH_RADIUS](#) = 6371000.

13.1.1 Typedef Documentation

13.1.1.1 `typedef boost::shared_ptr<FeltGridDefinition> felt::FeltGridDefinitionPtr`

13.1.1.2 `typedef short int felt::word`

A [felt](#) block "word" - 2 bytes.

13.1.2 Function Documentation

13.1.2.1 `std::ostream& felt::contentSummary (std::ostream & out, const FeltGridDefinition & grid)`

13.1.2.2 `template<typename T> T felt::get (word w) [inline]`

13.1.2.3 `boost::array<float, 6> felt::gridParameters (int gridType, int xNum, int yNum, int a, int b, int c, int d, const std::vector< short int > & extraData)`

retrieve the 6 gridparameters from the felt-data

Parameters:

gridType id of the grid, (header[8] < 1000) ? header[8] : (int) header[8] / 1000

xNum number of points in x-direction, header[9]

yNum number of points in x-direction, header[10]

a used for different depending on gridType, header[14]

b used for different depending on gridType, header[15]

c used for different depending on gridType, header[16]

d used for different depending on gridType, header[17]

extraData data at the end of the data-region, used for high resolution information (header[8] < 1000)
? 0 : header[8] % 1000

13.1.2.4 `std::string felt::gridParametersToProjDefinition (int gridType, const boost::array< float, 6 > & gridPars)`

convert the libmi-gridparameters to proj4 strings

Parameters:

gridType type defining the projection (1..6)

gridPars array containing libmi's six gridparameters

Returns:

proj.4 string

13.1.2.5 `bool felt::isUndefined (word w) [inline]`

Is the given word a "missing" value? This means that there exist no valid data for the given point

13.1.2.6 `boost::posix_time::ptime felt::parseTime (const word * data)`

13.1.2.7 `boost::posix_time::ptime felt::parseTimeNoThrow (const word * data)`

13.1.3 Variable Documentation

13.1.3.1 `const size_t felt::blockSize = blockWords * sizeof(word)`

13.1.3.2 `const size_t felt::blockWords = 1024`

13.1.3.3 `const double felt::EARTH_RADIUS = 6371000.`

earth-radius in m, taken from libmi rearth.f

13.1.3.4 `const size_t felt::offsetToContentDefinition = 2`

13.1.3.5 `const double felt::PI = M_PI`

mathematical pi

13.2 MetNoFelt Namespace Reference

Classes

- struct [ShortPairLess](#)
- class [Felt_Array](#)
encapsulate parameters of a [felt](#) file
- class [Felt_Array2](#)
A collection of [FeltFields](#) to build a 4-dimensional data array for 1-parameter and 1-vertical coordinate.
- class [Felt_File](#)
Felt File access.
- class [Felt_File2](#)
Felt File access.
- class [Felt_File_Error](#)
- struct [LevelPairLess](#)
- class [FeltParameters](#)

Typedefs

- typedef `set< pair< short, short >, ShortPairLess > ShortPairSet`
- typedef `map< pair< short, short >, short, ShortPairLess > ShortPairMap`
- typedef `std::pair< short, short > LevelPair`

Functions

- epoch_seconds [index16toTime](#) (const boost::array< short, 16 > &idx)
- `pair< short, short > index16toLevelPair (const boost::array< short, 16 > &idx)`
- `std::string getProjString (int gridType, const boost::array< float, 6 > &gridParameters) throw (Felt_File_Error)`
- const int [ANY_VALUE](#) ()
- const `std::string & UNDEFINED ()`
- const boost::array< short, 16 > & [ANY_ARRAY](#) ()
- const boost::array< short, 20 > & [ANY_ARRAY20](#) ()

13.2.1 Typedef Documentation

13.2.1.1 typedef std::pair<short, short> MetNoFelt::LevelPair

a pair with two level values

13.2.1.2 typedef map<pair<short,short>, short, ShortPairLess > MetNoFelt::ShortPairMap

map<pair<short,short>, short> with comparator

13.2.1.3 typedef set<pair<short,short>, ShortPairLess > MetNoFelt::ShortPairSet

set<pair<short,short> > with comparator

13.2.2 Function Documentation**13.2.2.1 const boost::array<short, 16>& MetNoFelt::ANY_ARRAY ()****13.2.2.2 const boost::array<short, 20>& MetNoFelt::ANY_ARRAY20 ()****13.2.2.3 const int MetNoFelt::ANY_VALUE () [inline]****13.2.2.4 std::string MetNoFelt::getProjString (int *gridType*, const boost::array< float, 6 > & *gridParameters*) throw (Felt_File_Error)****13.2.2.5 pair<short, short> MetNoFelt::index16toLevelPair (const boost::array< short, 16 > & *idx*)**

convert the 16-short header to a levelPair

13.2.2.6 epoch_seconds MetNoFelt::index16toTime (const boost::array< short, 16 > & *idx*)

convert the 16-short header to a time

13.2.2.7 const std::string& MetNoFelt::UNDEFINED ()

13.3 MetNoFimex Namespace Reference

Classes

- class [C_CDMReader](#)
- class [CachedForwardInterpolation](#)
- class [CachedInterpolationInterface](#)
- class [CachedInterpolation](#)
- class [CachedVectorReprojection](#)
- class [CDM](#)

Data structure of the Common Data Model.

- class [CDMAttribute](#)
- class [CDMDimension](#)
- class [CDMException](#)
- class [CDMExtractor](#)
- class [CDMFileReaderFactory](#)
- class [CDMInterpolator](#)
- class [CDMNamedEntity](#)
- struct [CDMNameCompare](#)
- class [CDMNameEqual](#)
- class [CDMNameEqualPtr](#)
- class [CDMQualityExtractor](#)

Extract data with defined quality status.

- class [CDMReader](#)

Basic interface for CDM reading and manipulation classes.

- class [CDMTimeInterpolator](#)
- class [CDMVariable](#)
- class [CDMWriter](#)
- class [CoordinateAxis](#)
- class [CoordinateSystem](#)
- struct [CompleteCoordinateSystemForComparator](#)
- class [LambertConformalConicProjection](#)
- class [LatitudeLongitudeProjection](#)
- class [MercatorProjection](#)
- class [PolarStereographicProjection](#)
- class [Projection](#)
- class [ProjectionImpl](#)
- class [RotatedLatitudeLongitudeProjection](#)
- class [StereographicProjection](#)
- class [Data](#)
- class [DataImpl](#)
- class [DataTypeChanger](#)
- class [FeltCDMReader](#)
- class [FeltCDMReader2](#)
- class [GribApiCDMWriter](#)
- class [GribApiCDMWriter_Impl1](#)
- class [GribApiCDMWriter_Impl2](#)

- class [GribApiCDMWriter_ImplAbstract](#)
- class [GribCDMReader](#)
- class [GribFileMessage](#)
- class [GribFileMessageEqualTime](#)

Functor to find Messages with equal time.

- class [GribFileMessageEqualLevelTime](#)

Functor to find messages with equal level and time.

- class [GribFileIndex](#)
- class [GridDefinition](#)
- class [Logger](#)
- class [NcmlCDMReader](#)
- class [NetCDF_CDMReader](#)
- class [NetCDF_CDMWriter](#)
- class [Null_CDMWriter](#)
- class [ReplaceStringObject](#)
- class [ReplaceStringTimeObject](#)
- class [SliceBuilder](#)
- class [SpatialAxisSpec](#)
- class [TimeLevelDataSliceFetcher](#)

read a slice of a given time/level combination from a cdmReader

- class [TimeSpec](#)
- class [FimexTime](#)
- class [TimeUnit](#)
- class [UnitException](#)
- class [Units](#)
- struct [staticCast](#)
- class [ScaleValue](#)
- class [ChangeMissingValue](#)
- class [XMLDoc](#)

Typedefs

- typedef boost::shared_ptr< [Logger](#) > [LoggerPtr](#)
- typedef long [epoch_seconds](#)
- typedef boost::shared_ptr< [xmlXPathObject](#) > [XPathObjPtr](#)

Enumerations

- enum [CDMDataType](#) {
[CDM_NAT](#) = 0, [CDM_CHAR](#), [CDM_SHORT](#), [CDM_INT](#),
[CDM_FLOAT](#), [CDM_DOUBLE](#), [CDM_STRING](#) }

Functions

- **DEPRECATED** (`std::vector< CDMAttribute > projStringToAttributes(std::string projStr)`)
convert a proj4 string to a list of CDMAttributes usable for CF-1.0 projection variable
- **DEPRECATED** (`std::string attributesToProjString(const std::vector< CDMAttribute > &attrs)`)
convert attributes of a projection-variable to a projString
- `CDMDataType string2datatype` (`const std::string &s`)
translate float/string/... to the appropriate CDMDataType
- `std::string datatype2string` (`CDMDataType type`)
- `std::ostream & operator<<` (`std::ostream &out, CoordinateAxis ca`)
- `std::ostream & operator<<` (`std::ostream &out, CoordinateAxis::AxisType t`)
- `std::ostream & operator<<` (`std::ostream &out, const CoordinateSystem &p`)
- `std::vector< boost::shared_ptr< const CoordinateSystem > > listCoordinateSystems` (`const CDM &cdm`)
- `std::ostream & operator<<` (`std::ostream &out, const Projection &proj`)
- `boost::shared_ptr< Data > createData` (`CDMDataType datatype, size_t length, double val=0`) throw (`CDMException`)
create a Data-pointer of the datatype
- `boost::shared_ptr< Data > createDataSlice` (`CDMDataType datatype, const Data &data, size_t dataStartPos, size_t dataSize`) throw (`CDMException`)
create a one-dimensional dataslice from another Data object
- `template<typename T1, typename T2>`
`boost::shared_array< T1 > duplicateArrayType` (`const boost::shared_array< T2 > &inData, long length`)
create a new shared array with a different type using static_cast
- `template<typename T1, typename T2>`
`const boost::shared_array< T1 > constConvertArrayType` (`const boost::shared_array< T2 > &inData, long length`)
return a shared array of this data, possibly pointer to internal data
- `template<class InputIterator>`
`boost::shared_ptr< Data > createData` (`CDMDataType datatype, InputIterator first, InputIterator last`) throw (`CDMException`)
create a Data-pointer of the datatype and fill with the data from the iterator
- `template<typename C>`
`void recursiveCopyMultiDimData` (`C **orgData, C **newData, const std::vector< size_t > &orgDimSize, const std::vector< size_t > &orgSliceSize, const std::vector< size_t > &newStart, const std::vector< size_t > &newSize, size_t currentDim`)
- `template<typename OUT, typename IN>`
`boost::shared_array< OUT > convertArrayType` (`const boost::shared_array< IN > &inData, size_t length, double oldFill, double oldScale, double oldOffset, double newFill, double newScale, double newOffset`)
- `boost::shared_ptr< FeltCDMReader2 > getFeltReader` (`std::string filename, std::string configFile-name`)

convenience function to retrieve a reader as shared_ptr instead of calling new

- `size_t gribDataRead` (const [GribFileMessage](#) &gfm, `std::vector< double >` &data, double missing-Value)
- `std::ostream & operator<<` (`std::ostream` &os, const [GribFileMessage](#) &gfm)
outputstream for a [GribFileMessage](#)
- `std::ostream & operator<<` (`std::ostream` &os, const [GribFileIndex](#) &gfm)
outputstream for a [GribFileIndex](#)
- `GridDefinition::Orientation gribGetGridOrientation` (boost::shared_ptr< [grib_handle](#) > gh)
- `Logger::LogLevel defaultLogLevel` ()
- `void defaultLogLevel` (Logger::LogLevel)
- `LoggerPtr getLogger` (const `std::string` &className)
- `std::ostream & operator<<` (`std::ostream` &out, const [FimexTime](#) &tTime)
- `FimexTime string2FimexTime` (const `std::string` &str) throw (CDMException)
- `void handleUdUnitError` (int unitErrCode, const `std::string` &message="") throw (UnitException)
- `int round` (double num)
- `std::string trim` (const `std::string` &str)
- `template<class InputIterator>`
`std::string join` (InputIterator start, InputIterator end, `std::string` delim=",")
- `template<class InputIterator>`
`std::string joinPtr` (InputIterator start, InputIterator end, `std::string` delim=",")
- `std::vector< std::string > tokenize` (const `std::string` &str, const `std::string` &delimiters=" ")
- `std::string string2lowerCase` (const `std::string` &str)
- `template<typename T>`
`std::string type2string` (T in)
- `template<>`
`std::string type2string< double >` (double in)
- `template<typename T>`
`T string2type` (`std::string` s)
- `epoch_seconds posixTime2epochTime` (const boost::posix_time::ptime &time)
- `template<typename T>`
`std::vector< T > tokenizeDotted` (const `std::string` &str, const `std::string` &delimiter=",") throw (CDMException)
- `std::string getXmlProp` (const [xmlNodePtr](#) node, const `std::string` &attrName)
- `std::string getXmlName` (const [xmlNodePtr](#) node)

13.3.1 Typedef Documentation

13.3.1.1 `typedef long MetNoFimex::epoch_seconds`

13.3.1.2 `typedef boost::shared_ptr<Logger> MetNoFimex::LoggerPtr`

13.3.1.3 `typedef boost::shared_ptr<xmlXPathObject> MetNoFimex::XPathObjPtr`

13.3.2 Enumeration Type Documentation

13.3.2.1 `enum MetNoFimex::CDMDataType`

Enumerator:

CDM_NAT

*CDM_CHAR**CDM_SHORT**CDM_INT**CDM_FLOAT**CDM_DOUBLE**CDM_STRING*

13.3.3 Function Documentation

13.3.3.1 `template<typename T1, typename T2> const boost::shared_array< T1 >
MetNoFimex::constConvertArrayType (const boost::shared_array< T2 > & inData,
long length) [inline]`

return a shared array of this data, possibly pointer to internal data

Parameters:

inData original data

length length of original data array

13.3.3.2 `template<typename OUT, typename IN> boost::shared_array<OUT>
MetNoFimex::convertArrayType (const boost::shared_array< IN > & inData, size_t
length, double oldFill, double oldScale, double oldOffset, double newFill, double
newScale, double newOffset) [inline]`

References `std::transform()`.

13.3.3.3 `template<class InputIterator> boost::shared_ptr< Data > MetNoFimex::createData
(CDMDataType datatype, InputIterator first, InputIterator last) throw (CDMException)
[inline]`

create a Data-pointer of the datatype and fill with the data from the iterator

Parameters:

datatype

first start of container containing the data to fill the array with

last end (excluded) of the container containing the data to fill the array with

Returns:

Base-Class ptr of the [DataImpl](#) belonging to the datatype

References `CDM_CHAR`, `CDM_DOUBLE`, `CDM_FLOAT`, `CDM_INT`, `CDM_NAT`, `CDM_SHORT`, and `std::distance()`.

13.3.3.4 `boost::shared_ptr<Data> MetNoFimex::createData (CDMDataType datatype, size_t length, double val = 0) throw (CDMException)`

create a Data-pointer of the datatype

Parameters:

datatype

size_t length of the data array

val default value for data elements, 0 by default

Returns:

Base-Class ptr of the [DataImpl](#) belonging to the datatype

13.3.3.5 `boost::shared_ptr<Data> MetNoFimex::createDataSlice (CDMDataType datatype, const Data & data, size_t dataStartPos, size_t dataSize) throw (CDMException)`

create a one-dimensional dataslice from another [Data](#) object

Parameters:

datatype of the return-data

data the data to read the values from, should be convertible data-format

dataStartPos the first element of data to fetch

dataSize the size of the data

13.3.3.6 `std::string MetNoFimex::datatype2string (CDMDataType type)`

13.3.3.7 `void MetNoFimex::defaultLogLevel (Logger::LogLevel)`

13.3.3.8 `Logger::LogLevel MetNoFimex::defaultLogLevel ()`

the defaultLogLevel can be used by the implemented logger to determine the minimum LogLevel. This value might be ignored/overwritten by a configuration within the implementation. It should be initialized in the main class.

13.3.3.9 `MetNoFimex::DEPRECATED (std::string attributesToProjString const std::vector< CDMAAttribute > &attrs)`

convert attributes of a projection-variable to a projString

[Deprecated](#)

use [Projection::create\(\)](#) with [Projection::getProj4String\(\)](#) instead

Parameters:

attrs attributes of the projection variable

Returns:

proj4 string

13.3.3.10 **MetNoFimex::DEPRECATED** (std::vector< CDMAttribute > projStringToAttributesstd::string projStr)

convert a proj4 string to a list of CDMAttributes usable for CF-1.0 projection variable
currently, projStrings of the form +proj=[stere] +lat_0=? +lon_0=? +lat_ts=?

Deprecated

use [Projection::createByProj4\(\)](#) and [Projection::getParameters\(\)](#)

13.3.3.11 **template**<typename T1, typename T2> boost::shared_array< T1 > MetNoFimex::duplicateArrayType (const boost::shared_array< T2 > & *inData*, long *length*) [inline]

create a new shared array with a different type using static_cast

Parameters:

inData original data

length length of original data array

References std::transform().

13.3.3.12 **boost::shared_ptr**<FeltCDMReader2> MetNoFimex::getFeltReader (std::string *filename*, std::string *configFilename*)

convenience function to retrieve a reader as shared_ptr instead of calling new

13.3.3.13 **LoggerPtr** MetNoFimex::getLogger (const std::string & *className*)

Retrieve a logger for Fimex. It will use loggers in the following order, skipping to the next one if the current one is not available: 1) log4cxx 2) no/dummy logger

13.3.3.14 **std::string** MetNoFimex::getXmlName (const xmlNodePtr *node*)

a memory-save form of xmlGetName

Returns:

a string of the attribute, "" if attribute doesn't exist

13.3.3.15 **std::string** MetNoFimex::getXmlProp (const xmlNodePtr *node*, const std::string & *attrName*)

a memory-save form of xmlGetProp

Returns:

a string of the attribute, "" if attribute doesn't exist

13.3.3.16 `size_t MetNoFimex::gribDataRead (const GribFileMessage & gfm, std::vector< double > & data, double missingValue)`

read the data corresponding to the *gfm* to the vector *data* of at maximum *data.size()* will be read.

Parameters:

gfm [GribFileMessage](#) to read from
data the storage the data will be read to
missingValue the missing- / fill-value the returned data will have

Returns:

the actual amount of data read

13.3.3.17 `GridDefinition::Orientation MetNoFimex::gribGetGridOrientation (boost::shared_ptr< grib_handle > gh)`

get the orientation of the data

Parameters:

gh grib-handle

13.3.3.18 `void MetNoFimex::handleUdUnitError (int unitErrCode, const std::string & message = " ") throw (UnitException)`

13.3.3.19 `template<class InputIterator> std::string MetNoFimex::join (InputIterator start, InputIterator end, std::string delim = ", ") [inline]`

Join values from an iterator to a string, using delimiter as separator.

Parameters:

start
end
delim separator, default to ",",

References `std::basic_ostringstream< _CharT, _Traits, _Alloc >::str()`.

13.3.3.20 `template<class InputIterator> std::string MetNoFimex::joinPtr (InputIterator start, InputIterator end, std::string delim = ", ") [inline]`

Join values from an iterator of pointers to a string, using delimiter as separator.

Parameters:

start
end
delim separator, default to ",",

References `std::basic_ostringstream< _CharT, _Traits, _Alloc >::str()`.

13.3.3.21 `std::vector<boost::shared_ptr<const CoordinateSystem> > MetNoFimex::listCoordinateSystems (const CDM & cdm)`

fetch all coordinate system from a [MetNoFimex::CDM](#)

Examples:

[coordinateSystem.cpp](#).

13.3.3.22 `std::ostream& MetNoFimex::operator<< (std::ostream & out, const FimexTime & fTime)`

13.3.3.23 `std::ostream& MetNoFimex::operator<< (std::ostream & os, const GribFileIndex & gfm)`

outputstream for a [GribFileIndex](#)

13.3.3.24 `std::ostream& MetNoFimex::operator<< (std::ostream & os, const GribFileMessage & gfm)`

outputstream for a [GribFileMessage](#)

13.3.3.25 `std::ostream& MetNoFimex::operator<< (std::ostream & out, const Projection & proj)`

output-stream for projections, implemented using toString()

13.3.3.26 `std::ostream& MetNoFimex::operator<< (std::ostream & out, const CoordinateSystem & p)`

output operator

13.3.3.27 `std::ostream& MetNoFimex::operator<< (std::ostream & out, CoordinateAxis::AxisType t)`

13.3.3.28 `std::ostream& MetNoFimex::operator<< (std::ostream & out, CoordinateAxis ca)`

13.3.3.29 `epoch_seconds MetNoFimex::posixTime2epochTime (const boost::posix_time::ptime & time)`

convert a posixTime to seconds since 1970-01-01

Parameters:

time time to convert

13.3.3.30 `template<typename C> void MetNoFimex::recursiveCopyMultiDimData (C **
orgData, C ** newData, const std::vector< size_t > & orgDimSize, const std::vector<
size_t > & orgSliceSize, const std::vector< size_t > & newStart, const std::vector<
size_t > & newSize, size_t currentDim) [inline]`

recursively copy data by moving the *newData* and *orgData* pointers forward and copy the data at the current position

it's assumed that the first dim in the vector is the fastest moving (fortran like)

Parameters:

orgData pointer to the current position of the original array

newData pointer to the current position of the new array the original dimensions of *orgData* helper-array with *orgSliceSize*[0] = 1; *orgSliceSize*[n] = *orgDimSize*[n] * *orgSliceSize*[n-1] the start positions in the new data the dimensions of the *newData* the dimension currently under work, should be between (*orgData*.size()-1) and 0

References `std::copy()`.

Referenced by `MetNoFimex::DataImpl< C >::slice()`.

13.3.3.31 `int MetNoFimex::round (double num)`

Round a double to integer.

13.3.3.32 `CDMDDataType MetNoFimex::string2datatype (const std::string & s)`

translate float/string/... to the appropriate CDMDDataType

13.3.3.33 `FimexTime MetNoFimex::string2FimexTime (const std::string & str) throw
(CDMEException)`

13.3.3.34 `std::string MetNoFimex::string2lowerCase (const std::string & str)`

convert a string to lowercase

13.3.3.35 `template<typename T> T MetNoFimex::string2type (std::string s) [inline]`

13.3.3.36 `std::vector<std::string> MetNoFimex::tokenize (const std::string & str, const
std::string & delimiters = " ")`

Tokenize a string by a delimiter. This function will automatically remove empty strings at the beginning or anywhere inside the string.

This function has been derived from <http://www.oopweb.com/CPP/Documents/CPPHOWTO/Volume/C++Programs/>

Parameters:

str the string to tokenize

delimiters the delimiters between the tokens

Returns:

vector of tokens

Referenced by tokenizeDotted().

13.3.3.37 `template<typename T> std::vector<T> MetNoFimex::tokenizeDotted (const std::string & str, const std::string & delimiter = " , ") throw (CDMException)` `[inline]`

convert a string with dots to a vector with type T

Parameters:

str f.e. 3.5,4.5,....,17.5

delimiter optional delimiter, defaults to ,

References std::vector< _Tp, _Alloc >::begin(), std::vector< _Tp, _Alloc >::end(), std::vector< _Tp, _Alloc >::push_back(), std::vector< _Tp, _Alloc >::size(), tokenize(), trim(), and type2string().

13.3.3.38 `std::string MetNoFimex::trim (const std::string & str)`

Remove leading and trailing spaces.

Parameters:

str string to trim

Referenced by tokenizeDotted().

13.3.3.39 `template<typename T> std::string MetNoFimex::type2string (T in)` `[inline]`

convert a type (i.e. int, float) to string representation

References std::basic_ostringstream< _CharT, _Traits, _Alloc >::str().

Referenced by MetNoFimex::DataImpl< C >::setValues(), MetNoFimex::DataImpl< C >::slice(), and tokenizeDotted().

13.3.3.40 `template<> std::string MetNoFimex::type2string< double > (double in)` `[inline]`

specialization for high prececision

Chapter 14

Class Documentation

14.1 `binary< N >` Struct Template Reference

```
#include <binaryConstants.h>
```

Public Types

- enum { `value` = (N % 8) + (binary<N/8>::value << 1) }

14.1.1 Detailed Description

`template<unsigned long long N> struct binary< N >`

use `binary<01001001>::value` as constant, works with up to 10 bits use `binary<01001001ULL>::value` as constant, works with up to 22 bits

Warning:

always start with leading 0, since all values have to be octals!!!

14.1.2 Member Enumeration Documentation

14.1.2.1 `template<unsigned long long N> anonymous enum`

Enumerator:

value

The documentation for this struct was generated from the following file:

- include/fimex/[binaryConstants.h](#)

14.2 `binary< 0 >` Struct Template Reference

```
#include <binaryConstants.h>
```

Public Types

- enum { `value` = 0 }

```
template<> struct binary< 0 >
```

14.2.1 Member Enumeration Documentation

14.2.1.1 anonymous enum

Enumerator:

value

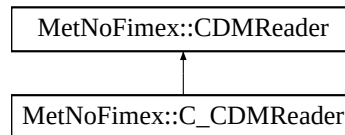
The documentation for this struct was generated from the following file:

- `include/fimex/binaryConstants.h`

14.3 MetNoFimex::C_CDMReader Class Reference

```
#include <C_CDMReader.h>
```

Inheritance diagram for MetNoFimex::C_CDMReader::



Public Member Functions

- [C_CDMReader](#) (boost::shared_ptr< [CDMReader](#) > dataReader)
- virtual [~C_CDMReader](#) ()
- virtual boost::shared_ptr< [Data](#) > [getDataSlice](#) (const **std::string** &varName, size_t unLimDimPos) throw (CDMException)
data-reading function to be called from the [CDMWriter](#)
- virtual void [setDoubleCallbackFunction](#) (const **std::string** &varName, [doubleDatasliceCallbackPtr](#) callback)

14.3.1 Detailed Description

This class should be used by people who want write an implementation of a [CDMReader](#) in C. They should set a callback-function to retrieve a variable with the [getDataSlice](#) functions.

14.3.2 Constructor & Destructor Documentation

14.3.2.1 MetNoFimex::C_CDMReader::C_CDMReader (boost::shared_ptr< [CDMReader](#) > dataReader)

14.3.2.2 virtual MetNoFimex::C_CDMReader::~~C_CDMReader () [virtual]

14.3.3 Member Function Documentation

14.3.3.1 virtual boost::shared_ptr<[Data](#)> MetNoFimex::C_CDMReader::getDataSlice (const **std::string** & varName, size_t unLimDimPos) throw (CDMException) [virtual]

data-reading function to be called from the [CDMWriter](#)

This methods needs to be implemented by the [CDMReader](#). It should provide the data for each variable, either by reading from disk, converting from another [CDMReader](#) or reading from an in-memory data-section.

This function should retrieve the whole data for a dataset without unlimited dimension if the unLimDimPos == 0.

Parameters:

varName name of the variable to read

unLimDimPos (optional) if the variable contains a unlimited dimension (max one allowed) an slice of this position is returned

Implements [MetNoFimex::CDMReader](#).

14.3.3.2 `virtual void MetNoFimex::C_CDMReader::setDoubleCallbackFunction (const std::string & varName, doubleDatasliceCallbackPtr callback)` [virtual]

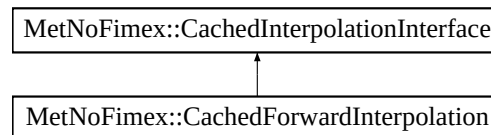
The documentation for this class was generated from the following file:

- [include/fimex/C_CDMReader.h](#)

14.4 MetNoFimex::CachedForwardInterpolation Class Reference

```
#include <CachedForwardInterpolation.h>
```

Inheritance diagram for MetNoFimex::CachedForwardInterpolation::



Public Member Functions

- [CachedForwardInterpolation](#) (int funcType, **std::vector**< double > pointsOnXAxis, **std::vector**< double > pointsOnYAxis, size_t inX, size_t inY, size_t outX, size_t outY)
- virtual [~CachedForwardInterpolation](#) ()
- virtual boost::shared_array< float > [interpolateValues](#) (boost::shared_array< float > inData, size_t size, size_t &newSize) const

14.4.1 Constructor & Destructor Documentation

14.4.1.1 MetNoFimex::CachedForwardInterpolation::CachedForwardInterpolation (int *funcType*, **std::vector**< double > *pointsOnXAxis*, **std::vector**< double > *pointsOnYAxis*, size_t *inX*, size_t *inY*, size_t *outX*, size_t *outY*)

14.4.1.2 virtual MetNoFimex::CachedForwardInterpolation::~~CachedForwardInterpolation ()
[inline, virtual]

14.4.2 Member Function Documentation

14.4.2.1 virtual boost::shared_array<float> MetNoFimex::CachedForwardInterpolation::interpolateValues (boost::shared_array< float > *inData*, size_t *size*, size_t & *newSize*) const [virtual]

Implements [MetNoFimex::CachedInterpolationInterface](#).

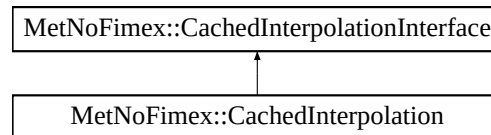
The documentation for this class was generated from the following file:

- include/fimex/[CachedForwardInterpolation.h](#)

14.5 MetNoFimex::CachedInterpolation Class Reference

```
#include <CachedInterpolation.h>
```

Inheritance diagram for MetNoFimex::CachedInterpolation::



Public Member Functions

- [CachedInterpolation](#) (int funcType, **std::vector**< double > pointsOnXAxis, **std::vector**< double > pointsOnYAxis, size_t inX, size_t inY, size_t outX, size_t outY)
- virtual [~CachedInterpolation](#) ()
- virtual boost::shared_array< float > [interpolateValues](#) (boost::shared_array< float > inData, size_t size, size_t &newSize) const

14.5.1 Detailed Description

Container to cache projection details to speed up interpolation of lots of fields.

14.5.2 Constructor & Destructor Documentation

14.5.2.1 MetNoFimex::CachedInterpolation::CachedInterpolation (int *funcType*, **std::vector**< double > *pointsOnXAxis*, **std::vector**< double > *pointsOnYAxis*, size_t *inX*, size_t *inY*, size_t *outX*, size_t *outY*)

Parameters:

funcType [interpolation.h](#) interpolation method

pointsOnXAxis projected values of the new projections coordinates expressed in the current x-coordinate (size = outX*outY)

pointsOnYAxis projected values of the new projections coordinates expressed in the current y-coordinate (size = outX*outY)

inX size of current X axis

inY size of current Y axis

outX size of new X axis

outY size of new Y axis

14.5.2.2 `virtual MetNoFimex::CachedInterpolation::~~CachedInterpolation ()` `[inline, virtual]`

14.5.3 Member Function Documentation

14.5.3.1 `virtual boost::shared_array<float> MetNoFimex::CachedInterpolation::interpolateValues (boost::shared_array< float > inData, size_t size, size_t & newSize) const` `[virtual]`

Actually interpolate the data. The data will be interpolated as floats internally.

Parameters:

inData the input data

the size of the input data array

newSize return the size of the output-array

Implements [MetNoFimex::CachedInterpolationInterface](#).

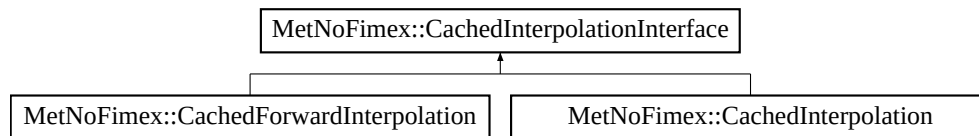
The documentation for this class was generated from the following file:

- [include/fimex/CachedInterpolation.h](#)

14.6 MetNoFimex::CachedInterpolationInterface Class Reference

```
#include <CachedInterpolation.h>
```

Inheritance diagram for MetNoFimex::CachedInterpolationInterface::



Public Member Functions

- virtual `boost::shared_array< float > interpolateValues` (`boost::shared_array< float > inData`, `size_t size`, `size_t & newSize`) `const =0`

14.6.1 Detailed Description

Interface for new cached spatial interpolation as used in [MetNoFimex::CDMInterpolator](#)

14.6.2 Member Function Documentation

- 14.6.2.1** `virtual boost::shared_array<float> MetNoFimex::CachedInterpolationInterface::interpolateValues`
`(boost::shared_array< float > inData, size_t size, size_t & newSize) const` [pure virtual]

Implemented in [MetNoFimex::CachedForwardInterpolation](#), and [MetNoFimex::CachedInterpolation](#).

The documentation for this class was generated from the following file:

- `include/fimex/CachedInterpolation.h`

14.7 MetNoFimex::CachedVectorReprojection Class Reference

```
#include <CachedVectorReprojection.h>
```

Public Member Functions

- [CachedVectorReprojection](#) ()
- [CachedVectorReprojection](#) (int *method*, boost::shared_array< double > *matrix*, int *ox*, int *oy*)
- virtual [~CachedVectorReprojection](#) ()
- void [reprojectValues](#) (boost::shared_array< float > &*uValues*, boost::shared_array< float > &*vValues*, size_t *size*) const throw (CDMException)
- size_t [getXSize](#) () const
- size_t [getYSize](#) () const

14.7.1 Constructor & Destructor Documentation

14.7.1.1 MetNoFimex::CachedVectorReprojection::CachedVectorReprojection () [inline]

14.7.1.2 MetNoFimex::CachedVectorReprojection::CachedVectorReprojection (int *method*, boost::shared_array< double > *matrix*, int *ox*, int *oy*) [inline]

14.7.1.3 virtual MetNoFimex::CachedVectorReprojection::~~CachedVectorReprojection () [inline, virtual]

14.7.2 Member Function Documentation

14.7.2.1 void MetNoFimex::CachedVectorReprojection::reprojectValues (boost::shared_array< float > & *uValues*, boost::shared_array< float > & *vValues*, size_t *size*) const throw (CDMException)

reproject the vector values

Parameters:

uValues the values in x-direction. These will be changed in-place.

vValues the values in y-direction. These will be changed in-place.

size the size of both arrays

14.7.2.2 size_t MetNoFimex::CachedVectorReprojection::getXSize () const [inline]

14.7.2.3 size_t MetNoFimex::CachedVectorReprojection::getYSize () const [inline]

The documentation for this class was generated from the following file:

- include/fimex/[CachedVectorReprojection.h](#)

14.8 MetNoFimex::CDM Class Reference

[Data](#) structure of the Common Data Model.

```
#include <fimex/CDM.h>
```

Public Types

- typedef `std::vector< CDMAttribute > AttrVec`
- typedef `std::map< std::string, AttrVec > StrAttrVecMap`
- typedef `std::vector< CDMDimension > DimVec`
- typedef `std::vector< CDMVariable > VarVec`

Public Member Functions

- [CDM](#) ()
- [CDM](#) (const [CDM](#) &rhs)
- virtual `~CDM` ()
- [CDM](#) & `operator=` (const [CDM](#) &rhs)
- void [addVariable](#) (const [CDMVariable](#) &var) throw (CDMException)
add variable to cdm
- [CDMVariable](#) & [getVariable](#) (const `std::string` &varName) throw (CDMException)
get a reference of a variable
- const [CDMVariable](#) & [getVariable](#) (const `std::string` &varName) const throw (CDMException)
get a reference of a variable
- bool [hasVariable](#) (const `std::string` &varName) const
test if variable exists
- `std::vector< std::string >` [findVariables](#) (const `std::string` &attrName, const `std::string` &attrValueRegEx) const
search for variable with certain attribute-value
- `std::vector< std::string >` [findVariables](#) (const `std::map< std::string, std::string >` &findAttributes, const `std::vector< std::string >` &findDimensions) const
search for variable with attribute-values and dimensions
- bool [renameVariable](#) (const `std::string` &oldName, const `std::string` &newName)
rename a variable
- bool [checkVariableAttribute](#) (const `std::string` &varName, const `std::string` &attribute, const `boost::regex` &attrValue) const
- void [removeVariable](#) (const `std::string` &variableName)
remove a variable and corresponding attributes
- void [addDimension](#) (const [CDMDimension](#) &dim) throw (CDMException)
add a dimension to cdm

- bool [hasDimension](#) (const **std::string** &dimName) const
- [CDMDimension](#) & [getDimension](#) (const **std::string** &dimName) throw (CDMException)
get a reference to a dimension
- const [CDMDimension](#) & [getDimension](#) (const **std::string** &dimName) const throw (CDMException)
- bool [testDimensionInUse](#) (const **std::string** &name) const
test if a dimension is actively in use
- bool [renameDimension](#) (const **std::string** &oldName, const **std::string** &newName) throw (CDMException)
rename a dimension
- bool [removeDimension](#) (const **std::string** &name) throw (CDMException)
remove a dimension
- const [CDMDimension](#) * [getUnlimitedDim](#) () const
retrieve the unlimited dimension
- bool [hasUnlimitedDim](#) (const [CDMVariable](#) &var) const
test if a variable contains the unlimited dim
- void [addAttribute](#) (const **std::string** &varName, const [CDMAttribute](#) &attr) throw (CDMException)
- void [addOrReplaceAttribute](#) (const **std::string** &varName, const [CDMAttribute](#) &attr) throw (CDMException)
- void [removeAttribute](#) (const **std::string** &varName, const **std::string** &attrName)
- void [toXMLStream](#) (**std::ostream** &os) const
print a xml representation to the stream
- const **DimVec** & [getDimensions](#) () const
get the dimension
- const **VarVec** & [getVariables](#) () const
get the variables
- const **StrAttrVecMap** & [getAttributes](#) () const
get the attributes
- **std::vector**< [CDMAttribute](#) > [getAttributes](#) (const **std::string** &varName) const
get the attributes of an variable
- [CDMAttribute](#) & [getAttribute](#) (const **std::string** &varName, const **std::string** &attrName) throw (CDMException)
get an attribute
- const [CDMAttribute](#) & [getAttribute](#) (const **std::string** &varName, const **std::string** &attrName) const throw (CDMException)
get a const. attribute
- bool [getAttribute](#) (const **std::string** &varName, const **std::string** &attrName, [CDMAttribute](#) &retAttribute) const

get an attribute without throwing an error

- double [getFillValue](#) (const **std::string** &varName) const
- **std::string** [getUnits](#) (const **std::string** &varName) const
- void [generateProjectionCoordinates](#) (const **std::string** &projectionVariable, const **std::string** &xDim, const **std::string** &yDim, const **std::string** &lonDim, const **std::string** &latDim) throw (CDMException)

generate the projection coordinates (usually named "lat lon")

- **DEPRECATED** (bool [getProjectionAndAxesUnits](#)(**std::string** &projectionName, **std::string** &xAxis, **std::string** &yAxis, **std::string** &xAxisUnits, **std::string** &yAxisUnits) const throw(CDMException))

extract the names of the projection-variable and the corresponding projection-axes

- **DEPRECATED** (**AttrVec** [getProjection](#)(**std::string** varName) const)

get the projection attributes (as of CF-1.0) of a variable

- boost::shared_ptr< const [Projection](#) > [getProjectionOf](#) (**std::string** varName) const

get the projection of a variable

- **std::string** [getHorizontalXAxis](#) (**std::string** varName) const

get the x-(lon) axis of the variable

- **std::string** [getHorizontalYAxis](#) (**std::string** varName) const

get the y-(lat) axis of the variable

- bool [getLatitudeLongitude](#) (**std::string** varName, **std::string** &latitude, **std::string** &longitude) const

detect the latitude and longitude coordinates of the variable

- **std::string** [getTimeAxis](#) (**std::string** varName) const

get the time axis of the variable

- **std::string** [getVerticalAxis](#) (**std::string** varName) const

get the vertical axis of the variable

Static Public Member Functions

- static const **std::string** & [globalAttributeNS](#) ()

the namespace for global attributes

14.8.1 Detailed Description

[Data](#) structure of the Common Data Model.

This class implements the data-structure of the Common Data Model version 1
<http://www.unidata.ucar.edu/software/netcdf-java/CDM.html>

14.8.2 Member Typedef Documentation

14.8.2.1 `typedef std::vector<CDMAttribute> MetNoFimex::CDM::AttrVec`

14.8.2.2 `typedef std::map<std::string, AttrVec> MetNoFimex::CDM::StrAttrVecMap`

14.8.2.3 `typedef std::vector<CDMDimension> MetNoFimex::CDM::DimVec`

14.8.2.4 `typedef std::vector<CDMVariable> MetNoFimex::CDM::VarVec`

14.8.3 Constructor & Destructor Documentation

14.8.3.1 `MetNoFimex::CDM::CDM ()`

14.8.3.2 `MetNoFimex::CDM::CDM (const CDM & rhs)`

14.8.3.3 `virtual MetNoFimex::CDM::~~CDM () [virtual]`

14.8.4 Member Function Documentation

14.8.4.1 `CDM& MetNoFimex::CDM::operator= (const CDM & rhs)`

14.8.4.2 `void MetNoFimex::CDM::addVariable (const CDMVariable & var) throw (CDMException)`

add variable to cdm

Parameters:

var the variable to add

Exceptions:

CDMException if `var.varName()` already exists

14.8.4.3 `CDMVariable& MetNoFimex::CDM::getVariable (const std::string & varName) throw (CDMException)`

get a reference of a variable

Parameters:

varName name of the variable

Exceptions:

CDMException if `varName` doesn't exist

14.8.4.4 `const CDMVariable& MetNoFimex::CDM::getVariable (const std::string & varName) const throw (CDMException)`

get a reference of a variable

this is a constant version of `CDMVariable::getVariable`

Parameters:

varName name of the variable

Exceptions:

CDMException if *varName* doesn't exist

14.8.4.5 bool MetNoFimex::CDM::hasVariable (const std::string & *varName*) const

test if variable exists

Parameters:

varName name of variable

14.8.4.6 std::vector<std::string> MetNoFimex::CDM::findVariables (const std::string & *attrName*, const std::string & *attrValueRegExp*) const

search for variable with certain attribute-value

Parameters:

attrName name of the attribute

attrValueRegExp regular expression the 'string'-value needs to match

Returns:

variable names of the variable with attributes matching

14.8.4.7 std::vector<std::string> MetNoFimex::CDM::findVariables (const std::map<std::string, std::string> & *findAttributes*, const std::vector<std::string> & *findDimensions*) const

search for variable with attribute-values and dimensions

And AND search for attributes and dimensions.

Parameters:

findAttributes map with (attribute => string-value regExp) pairs

findDimensions vector with dimensions contained in variable

Returns:

variable names of the variable with attributes matching the request and containing all dimensions

14.8.4.8 bool MetNoFimex::CDM::renameVariable (const std::string & *oldName*, const std::string & *newName*)

rename a variable

Parameters:

oldName the old name of the variable

newName the new name of the variable

Returns:

1 on success (*oldName* exists), 0 on failure

Warning:

this will not change the spatialVectorCounterPart of all other variables

14.8.4.9 bool MetNoFimex::CDM::checkVariableAttribute (const std::string & *varName*, const std::string & *attribute*, const boost::regex & *attrValue*) const

check if a variable contains a attributes with a matching string-value

Parameters:

varName variable

attribute the attribute name

attrValue the regexp the string-value of the attribute will match against

14.8.4.10 void MetNoFimex::CDM::removeVariable (const std::string & *variableName*)

remove a variable and corresponding attributes

Parameters:

variableName the variable to remove

14.8.4.11 void MetNoFimex::CDM::addDimension (const CDMDimension & *dim*) throw (CDMException)

add a dimension to cdm

Parameters:

dim the dimension

Exceptions:

[*CDMException*](#) if *dim*-name already exists

14.8.4.12 bool MetNoFimex::CDM::hasDimension (const std::string & *dimName*) const

check if the dimension exists

Parameters:

dimName name of the dimension

14.8.4.13 CDMDimension& MetNoFimex::CDM::getDimension (const std::string & *dimName*) throw (CDMException)

get a reference to a dimension

Parameters:

dimName name of the dimension

Exceptions:

[*CDMException*](#) if dimension doesn't exist

14.8.4.14 const CDMDimension& MetNoFimex::CDM::getDimension (const std::string & *dimName*) const throw (CDMException)**14.8.4.15 bool MetNoFimex::CDM::testDimensionInUse (const std::string & *name*) const**

test if a dimension is actively in use

Parameters:

name dimensionName

14.8.4.16 bool MetNoFimex::CDM::renameDimension (const std::string & *oldName*, const std::string & *newName*) throw (CDMException)

rename a dimension

Rename a dimension.

Returns:

false if the original name does not exist.

Exceptions:

[*CDMException*](#) if newName already in use in a variable but for a different dimension

14.8.4.17 bool MetNoFimex::CDM::removeDimension (const std::string & *name*) throw (CDMException)

remove a dimension

Remove a dimension, if it is not in use by a variable.

Returns:

true if dimension existed, false otherwise

Exceptions:

[*CDMException*](#) if dimension in us in a variable

14.8.4.18 const CDMDimension* MetNoFimex::CDM::getUnlimitedDim () const

retrieve the unlimited dimension

Returns:

unLimDim pointer with the unlimited dimension, the pointer will be deleted with the [CDM](#)

14.8.4.19 bool MetNoFimex::CDM::hasUnlimitedDim (const CDMVariable & var) const

test if a variable contains the unlimited dim

Returns:

true/false

14.8.4.20 void MetNoFimex::CDM::addAttribute (const std::string & varName, const CDMAttribute & attr) throw (CDMException)

add an attribute to cdm

Parameters:

varName name of the variabl the attribute belongs to

attr the [CDMAttribute](#)

Exceptions:

[*CDMException*](#) if varName doesn't exist, or attr.getName() already exists

14.8.4.21 void MetNoFimex::CDM::addOrReplaceAttribute (const std::string & varName, const CDMAttribute & attr) throw (CDMException)

add or replace an attribute of the cdm

Parameters:

varName name of variable the attribute belongs to

attr the [CDMAttribute](#)

Exceptions:

[*CDMException*](#) if vaName doesn't exist

14.8.4.22 `void MetNoFimex::CDM::removeAttribute (const std::string & varName, const std::string & attrName)`

remove an attribute from the cdm

Parameters:

varName name of variable the attribute belongs to

attr the [CDMAttribute](#)

14.8.4.23 `void MetNoFimex::CDM::toXMLStream (std::ostream & os) const`

print a xml representation to the stream

14.8.4.24 `static const std::string& MetNoFimex::CDM::globalAttributeNS () [inline, static]`

the namespace for global attributes

14.8.4.25 `const DimVec& MetNoFimex::CDM::getDimensions () const`

get the dimension

14.8.4.26 `const VarVec& MetNoFimex::CDM::getVariables () const`

get the variables

14.8.4.27 `const StrAttrVecMap& MetNoFimex::CDM::getAttributes () const`

get the attributes

Returns:

map of type <variableName <attributeName, attribute>>

14.8.4.28 `std::vector<CDMAttribute> MetNoFimex::CDM::getAttributes (const std::string & varName) const`

get the attributes of an variable

Parameters:

varName name of variable

14.8.4.29 CDMAttribute& MetNoFimex::CDM::getAttribute (const std::string & *varName*, const std::string & *attrName*) throw (CDMException)

get an attribute

Parameters:

varName name of variable

attrName name of attribute

Exceptions:

CDMException if *varName* *attrName* combination doesn't exists

14.8.4.30 const CDMAttribute& MetNoFimex::CDM::getAttribute (const std::string & *varName*, const std::string & *attrName*) const throw (CDMException)

get a const. attribute

Parameters:

varName name of variable

attrName name of attribute

Exceptions:

CDMException if *varName* *attrName* combination doesn't exists

14.8.4.31 bool MetNoFimex::CDM::getAttribute (const std::string & *varName*, const std::string & *attrName*, CDMAttribute & *retAttribute*) const

get an attribute without throwing an error

This method will search for an attribute in the cdm. It will return true on success and return the attribute.

Parameters:

varName name of variable

attrName name of attribute

retAttribute returns the attribute if found

Returns:

true when attribute has been found and set

14.8.4.32 double MetNoFimex::CDM::getFillValue (const std::string & *varName*) const

get the fill value of an variable (_FillValue attribute)

Returns:

value of _FillValue attribute, or MIFI_UNDEFINED_F

14.8.4.33 `std::string MetNoFimex::CDM::getUnits (const std::string & varName) const`

get the value of the "units" attribute

Returns:

unitsString or ""

14.8.4.34 `void MetNoFimex::CDM::generateProjectionCoordinates (const std::string & projectionVariable, const std::string & xDim, const std::string & yDim, const std::string & lonDim, const std::string & latDim) throw (CDMException)`

generate the projection coordinates (usually named "lat lon")

Parameters:

projectionVariable the variable containing the projection information

xDim the x dimension (the corresponding variable needs to contain data and units)

yDim the y dimension (the corresponding variable needs to contain data and units)

lonDim name of the longitude variable

latDim name of the latitude variable

Exceptions:

CDMException if any information is missing

14.8.4.35 `MetNoFimex::CDM::DEPRECATED (bool getProjectionAndAxesUnits std::string & projectionName, std::string & xAxis, std::string & yAxis, std::string & xAxisUnits, std::string & yAxisUnits) const throw (CDMException)`

extract the names of the projection-variable and the corresponding projection-axes

Parameters:

projectionName output of the projection variables name

xAxis output of the spatial x axis

yAxis output of the spation y axis

xAxisUnit output of unit for x axis

yAxisUnit output of unit for y axis

Returns:

true if unique result, false (and print warning) if results are not unique

Exceptions:

CDMException if no projection with corresponding axes can be found

14.8.4.36 MetNoFimex::CDM::DEPRECATED (AttrVec getProjection(std::string varName) const)

get the projection attributes (as of CF-1.0) of a variable

Parameters:

varName name of variable

Returns:

vector of attributes of the projection, an empty vector if no projection found

Deprecated

use the [getProjectionOf\(\)](#) method

14.8.4.37 boost::shared_ptr<const Projection> MetNoFimex::CDM::getProjectionOf (std::string varName) const

get the projection of a variable

This is the same as using the [CoordinateSystem::getProjection\(\)](#).

Parameters:

varName name of variable

Returns:

projection

14.8.4.38 std::string MetNoFimex::CDM::getHorizontalXAxis (std::string varName) const

get the x-(lon) axis of the variable

This is the same as using the [CoordinateSystem::getGeoXAxis\(\)](#).

Parameters:

varName name of variable

Returns:

name of x-axis dimension (or "" if not defined)

14.8.4.39 std::string MetNoFimex::CDM::getHorizontalYAxis (std::string varName) const

get the y-(lat) axis of the variable

This is the same as using the [CoordinateSystem::getGeoYAxis\(\)](#).

Parameters:

varName name of variable

Returns:

name of y-axis dimension (or "" if not defined)

14.8.4.40 **bool MetNoFimex::CDM::getLatitudeLongitude (std::string *varName*, std::string & *latitude*, std::string & *longitude*) const**

detect the latitude and longitude coordinates of the variable

This is the same as using the [CoordinateSystem::findAxisOfType\(\)](#) with [CoordinateAxis::Lon](#) and [CoordinateAxis::Lat](#).

Parameters:

varName name of variable

latitude return value of the latitude

longitude return value of the longitude

Returns:

true if latitude and longitude have been found

14.8.4.41 **std::string MetNoFimex::CDM::getTimeAxis (std::string *varName*) const**

get the time axis of the variable

This is the same as using the [CoordinateSystem::getTimeAxis\(\)](#).

Parameters:

varName name of variable

Returns:

name of time dimension (or "" if not defined)

14.8.4.42 **std::string MetNoFimex::CDM::getVerticalAxis (std::string *varName*) const**

get the vertical axis of the variable

This is the same as using the [CoordinateSystem::getGeoZAxis\(\)](#).

Parameters:

varName name of variable

Returns:

name of vertical dimension (or "" if not defined)

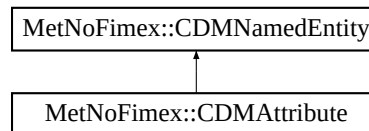
The documentation for this class was generated from the following file:

- include/fimex/[CDM.h](#)

14.9 MetNoFimex::CDMAttribute Class Reference

```
#include <CDMAttribute.h>
```

Inheritance diagram for MetNoFimex::CDMAttribute::



Public Member Functions

- [CDMAttribute](#) ()
- [CDMAttribute](#) (**std::string** name, **std::string** value)
create a string attribute
- [CDMAttribute](#) (**std::string** name, char value)
create a char attribute with a char array of length 1
- [CDMAttribute](#) (**std::string** name, int value)
create a int attribute with a int array of length 1
- [CDMAttribute](#) (**std::string** name, short value)
create a short attribute with a short array of length 1
- [CDMAttribute](#) (**std::string** name, float value)
create a float attribute with a float array of length 1
- [CDMAttribute](#) (**std::string** name, double value)
create a double attribute with a double array of length 1
- [CDMAttribute](#) (**std::string** name, [CDMDataType](#) datatype, boost::shared_ptr< [Data](#) > data)
create a attribute with the low level information
- [CDMAttribute](#) (const **std::string** &name, const **std::string** &datatype, const **std::string** &value) throw (CDMException)
create a attribute from a string representation
- [CDMAttribute](#) (const **std::string** &name, [CDMDataType](#) datatype, const **std::vector**< **std::string** > &values) throw (CDMException)
create a attribute with a vector of values in string representation
- virtual [~CDMAttribute](#) ()
- const **std::string** & [getName](#) () const
retrieve the name of the attribute
- void [setName](#) (**std::string** newName)
set the name of the attribute

- const **std::string** [getStringValue](#) () const
retrieve the stringified value of the attribute
- const boost::shared_ptr< [Data](#) > [getData](#) () const
retrieve the data-pointer of the attribute
- void [setData](#) (boost::shared_ptr< [Data](#) > data)
set the data for this attribute
- const [CDMDataType](#) [getDataType](#) () const
retrieve the datatype of the attribute
- void [toXMLStream](#) (**std::ostream** &out) const

14.9.1 Constructor & Destructor Documentation

14.9.1.1 MetNoFimex::CDMAAttribute::CDMAAttribute ()

14.9.1.2 MetNoFimex::CDMAAttribute::CDMAAttribute (std::string *name*, std::string *value*) [explicit]

create a string attribute

14.9.1.3 MetNoFimex::CDMAAttribute::CDMAAttribute (std::string *name*, char *value*) [explicit]

create a char attribute with a char array of length 1

14.9.1.4 MetNoFimex::CDMAAttribute::CDMAAttribute (std::string *name*, int *value*) [explicit]

create a int attribute with a int array of length 1

14.9.1.5 MetNoFimex::CDMAAttribute::CDMAAttribute (std::string *name*, short *value*) [explicit]

create a short attribute with a short array of length 1

14.9.1.6 MetNoFimex::CDMAAttribute::CDMAAttribute (std::string *name*, float *value*) [explicit]

create a float attribute with a float array of length 1

14.9.1.7 MetNoFimex::CDMAAttribute::CDMAAttribute (std::string *name*, double *value*) [explicit]

create a double attribute with a double array of length 1

14.9.1.8 MetNoFimex::CDMAAttribute::CDMAAttribute (std::string *name*, CDMDatatype *datatype*, boost::shared_ptr< Data > *data*) [explicit]

create a attribute with the low level information

14.9.1.9 MetNoFimex::CDMAAttribute::CDMAAttribute (const std::string & *name*, const std::string & *datatype*, const std::string & *value*) throw (CDMException) [explicit]

create a attribute from a string representation

14.9.1.10 MetNoFimex::CDMAAttribute::CDMAAttribute (const std::string & *name*, CDMDatatype *datatype*, const std::vector< std::string > & *values*) throw (CDMException) [explicit]

create a attribute with a vector of values in string representation

14.9.1.11 virtual MetNoFimex::CDMAAttribute::~~CDMAAttribute () [virtual]

14.9.2 Member Function Documentation

14.9.2.1 const std::string& MetNoFimex::CDMAAttribute::getName () const [inline, virtual]

retrieve the name of the attribute

Implements [MetNoFimex::CDMNamedEntity](#).

14.9.2.2 void MetNoFimex::CDMAAttribute::setName (std::string *newName*) [inline]

set the name of the attribute

14.9.2.3 const std::string MetNoFimex::CDMAAttribute::getStringValue () const

retrieve the stringified value of the attribute

14.9.2.4 const boost::shared_ptr<Data> MetNoFimex::CDMAAttribute::getData () const [inline]

retrieve the data-pointer of the attribute

14.9.2.5 void MetNoFimex::CDMAAttribute::setData (boost::shared_ptr< Data > *data*) [inline]

set the data for this attribute

14.9.2.6 const CDMDatatype MetNoFimex::CDMAAttribute::getDataType () const [inline]

retrieve the datatype of the attribute

14.9.2.7 void MetNoFimex::CDMAttribute::toXMLStream (std::ostream & *out*) const

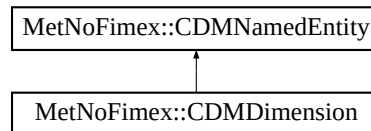
The documentation for this class was generated from the following file:

- include/fimex/[CDMAttribute.h](#)

14.10 MetNoFimex::CDMDimension Class Reference

```
#include <CDMDimension.h>
```

Inheritance diagram for MetNoFimex::CDMDimension::



Public Member Functions

- [CDMDimension \(\)](#)
- [CDMDimension \(std::string name, long length\)](#)
- virtual [~CDMDimension \(\)](#)
- const [std::string & getName \(\)](#) const
- void [setName \(std::string newName\)](#)
- [size_t getLength \(\)](#) const
- void [setLength \(size_t length\)](#)
- void [setUnlimited \(int unlimited\)](#)
- int [isUnlimited \(\)](#) const
- void [toXMLStream \(std::ostream &out\)](#) const

print xml representation to stream

14.10.1 Constructor & Destructor Documentation

14.10.1.1 MetNoFimex::CDMDimension::CDMDimension ()

14.10.1.2 MetNoFimex::CDMDimension::CDMDimension (std::string *name*, long *length*)

14.10.1.3 virtual MetNoFimex::CDMDimension::~~CDMDimension () [virtual]

14.10.2 Member Function Documentation

14.10.2.1 const std::string& MetNoFimex::CDMDimension::getName () const [inline, virtual]

Implements [MetNoFimex::CDMNamedEntity](#).

14.10.2.2 void MetNoFimex::CDMDimension::setName (std::string *newName*) [inline]

14.10.2.3 size_t MetNoFimex::CDMDimension::getLength () const [inline]

14.10.2.4 void MetNoFimex::CDMDimension::setLength (size_t *length*) [inline]

14.10.2.5 void MetNoFimex::CDMDimension::setUnlimited (int *unlimited*) [inline]

14.10.2.6 int MetNoFimex::CDMDimension::isUnlimited () const [inline]

14.10.2.7 void MetNoFimex::CDMDimension::toXMLStream (std::ostream & *out*) const

print xml representation to stream

Parameters:

out stream to write xml to

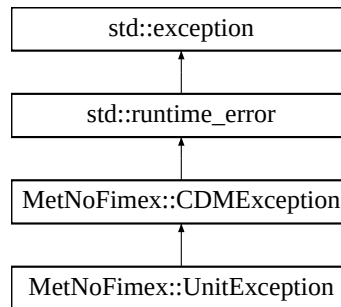
The documentation for this class was generated from the following file:

- include/fimex/[CDMDimension.h](#)

14.11 MetNoFimex::CDMException Class Reference

```
#include <CDMException.h>
```

Inheritance diagram for MetNoFimex::CDMException::



Public Member Functions

- [CDMException \(\)](#)
- [CDMException \(std::string msg\)](#)

14.11.1 Constructor & Destructor Documentation

14.11.1.1 MetNoFimex::CDMException::CDMException () [inline]

14.11.1.2 MetNoFimex::CDMException::CDMException (std::string msg) [inline, explicit]

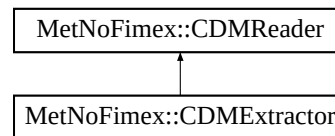
The documentation for this class was generated from the following file:

- include/fimex/[CDMException.h](#)

14.12 MetNoFimex::CDMExtractor Class Reference

```
#include <CDMExtractor.h>
```

Inheritance diagram for MetNoFimex::CDMExtractor::



Public Member Functions

- [CDMExtractor](#) (boost::shared_ptr< [CDMReader](#) > dataReader)
- virtual [~CDMExtractor](#) ()
- virtual boost::shared_ptr< [Data](#) > [getDataSlice](#) (const **std::string** &varName, size_t unLimDimPos=0) throw (CDMException)
data-reading function to be called from the [CDMWriter](#)
- virtual void [removeVariable](#) (**std::string** variable) throw (CDMException)
Remove a variable from the [CDM](#).
- virtual void [reduceDimension](#) (**std::string** dimName, size_t start, size_t length) throw (CDMException)
Reduce a dimension of the file.
- virtual void [reduceDimensionStartEnd](#) (**std::string** dimName, size_t start=0, long end=0) throw (CDMException)
Reduce a dimension of the file.
- virtual void [changeDataType](#) (**std::string** variable, [CDMDataType](#) datatype) throw (CDMException)
change the datatype of the variable

14.12.1 Constructor & Destructor Documentation

14.12.1.1 [MetNoFimex::CDMExtractor::CDMExtractor](#) (boost::shared_ptr< [CDMReader](#) > dataReader)

14.12.1.2 virtual [MetNoFimex::CDMExtractor::~~CDMExtractor](#) () [virtual]

14.12.2 Member Function Documentation

14.12.2.1 virtual boost::shared_ptr<[Data](#)> [MetNoFimex::CDMExtractor::getDataSlice](#) (const **std::string** & varName, size_t unLimDimPos = 0) throw (CDMException) [virtual]

data-reading function to be called from the [CDMWriter](#)

This methods needs to be implemented by the [CDMReader](#). It should provide the data for each variable, either by reading from disk, converting from another [CDMReader](#) or reading from an in-memory data-section.

This function should retrieve the whole data for a dataset without unlimited dimension if the `unLimDimPos == 0`.

Parameters:

varName name of the variable to read

unLimDimPos (optional) if the variable contains a unlimited dimension (max one allowed) an slice of this position is returned

Implements [MetNoFimex::CDMReader](#).

14.12.2.2 virtual void MetNoFimex::CDMExtractor::removeVariable (std::string *variable*) throw (CDMException) [virtual]

Remove a variable from the [CDM](#).

Parameters:

name of the variable

Exceptions:

[CDMException](#) if variable doesn't exist

14.12.2.3 virtual void MetNoFimex::CDMExtractor::reduceDimension (std::string *dimName*, size_t *start*, size_t *length*) throw (CDMException) [virtual]

Reduce a dimension of the file.

Parameters:

name dimension to change

start start-position corresponding to the original dimension

size size of the new dimension

Exceptions:

[CDMException](#) if dimension doesn't exist or start+size outside range of the original dimension

14.12.2.4 virtual void MetNoFimex::CDMExtractor::reduceDimensionStartEnd (std::string *dimName*, size_t *start* = 0, long *end* = 0) throw (CDMException) [virtual]

Reduce a dimension of the file.

Parameters:

name dimension to change

start start-position corresponding to the original dimension, defaults to 0

end end-position of dimension, 0 means full size, negative values start from end

Exceptions:

CDMException if dimension doesn't exist or start+size outside range of the original dimension

14.12.2.5 `virtual void MetNoFimex::CDMExtractor::changeDataType (std::string variable,
CDMDataType datatype) throw (CDMException) [virtual]`

change the datatype of the variable

a change of the variable will also change the datatype of the `_FillValue` attribute

Parameters:

variable name of the variable

datatype new datatype

Exceptions:

CDMException if variable doesn't exist or conversion to datatype is not supported

The documentation for this class was generated from the following file:

- `include/fimex/CDMExtractor.h`

14.13 MetNoFimex::CDMFileReaderFactory Class Reference

```
#include <CDMFileReaderFactory.h>
```

Static Public Member Functions

- static int [detectFileType](#) (const **std::string** &fileName)
detect the filetype of a input-file
- static boost::shared_ptr< [CDMReader](#) > [create](#) (int fileType, const **std::string** &fileName, const **std::string** &configFile, const **std::vector**< **std::string** > &args)
Factory for [CDMReader](#) of input-files.

14.13.1 Detailed Description

helper class to simplify file-reader detection and creation

14.13.2 Member Function Documentation

14.13.2.1 static int MetNoFimex::CDMFileReaderFactory::detectFileType (const **std::string** & *fileName*) [static]

detect the filetype of a input-file

The detectFileType function uses heuristics (appendix, magic characters) to detect the filetype

Parameters:

fileName input file

Returns:

one of the MIFI_FILETYPE_* flags

Exceptions:

if file not found

14.13.2.2 static boost::shared_ptr<CDMReader> MetNoFimex::CDMFileReaderFactory::create (int *fileType*, const **std::string** & *fileName*, const **std::string** & *configFile*, const **std::vector**< **std::string** > & *args*) [static]

Factory for [CDMReader](#) of input-files.

The function create reader tries to create a reader by filetype MIFI_FILETYPE_*. The optional arguments are defined by the different readers. Use default objects (empty string, empty vector) if arguments are not desired.

Parameters:

fileType, one of MIFI_FILETYPE_*, possibly read by [detectFileType\(\)](#)

fileName,name of input type

configFile

options optional options for the [CDMReader](#)

Returns:

pointer to [CDMReader](#)

Exceptions:

[CDMException](#) if type not compiled in, or creation fails

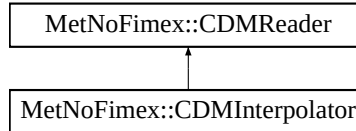
The documentation for this class was generated from the following file:

- [include/fimex/CDMFileReaderFactory.h](#)

14.14 MetNoFimex::CDMInterpolator Class Reference

```
#include <CDMInterpolator.h>
```

Inheritance diagram for MetNoFimex::CDMInterpolator::



Public Member Functions

- [CDMInterpolator](#) (boost::shared_ptr< [CDMReader](#) > dataReader)
- virtual [~CDMInterpolator](#) ()
- virtual boost::shared_ptr< [Data](#) > [getDataSlice](#) (const **std::string** &varName, size_t unLimDimPos=0) throw (CDMException)

retrieve data from the underlying dataReader and interpolate the values due to the current projection

- virtual void [changeProjection](#) (int method, const **std::string** &proj_input, const **std::vector**< double > &out_x_axis, const **std::vector**< double > &out_y_axis, const **std::string** &out_x_axis_unit, const **std::string** &out_y_axis_unit) throw (CDMException)
- virtual void [changeProjection](#) (int method, const **std::string** &proj_input, const **std::string** &out_x_axis, const **std::string** &out_y_axis, const **std::string** &out_x_axis_unit, const **std::string** &out_y_axis_unit) throw (CDMException)
- virtual void [setLatitudeName](#) (const **std::string** &latName)
- virtual const **std::string** & [getLatitudeName](#) () const
- virtual void [setLongitudeName](#) (const **std::string** &lonName)
- virtual const **std::string** & [getLongitudeName](#) () const

14.14.1 Constructor & Destructor Documentation

14.14.1.1 MetNoFimex::CDMInterpolator::CDMInterpolator (boost::shared_ptr< [CDMReader](#) > dataReader)

14.14.1.2 virtual MetNoFimex::CDMInterpolator::~~CDMInterpolator () [virtual]

14.14.2 Member Function Documentation

14.14.2.1 virtual boost::shared_ptr<[Data](#)> MetNoFimex::CDMInterpolator::getDataSlice (const **std::string** & varName, size_t unLimDimPos = 0) throw (CDMException) [virtual]

retrieve data from the underlying dataReader and interpolate the values due to the current projection

Implements [MetNoFimex::CDMReader](#).

14.14.2.2 `virtual void MetNoFimex::CDMInterpolator::changeProjection (int method, const std::string & proj_input, const std::vector< double > & out_x_axis, const std::vector< double > & out_y_axis, const std::string & out_x_axis_unit, const std::string & out_y_axis_unit) throw (CDMException) [virtual]`

@ brief change the (main) projection of the dataReaders cdm to this new projection

Parameters:

method Interpolation method
proj_input input-string for proj4, used as output projection
out_x_axis values of the output x-axis
out_y_axis values of the output y-axis
out_x_axis_unit unit of the output x-axis
out_y_axis_unit unit of the output y-axis

14.14.2.3 `virtual void MetNoFimex::CDMInterpolator::changeProjection (int method, const std::string & proj_input, const std::string & out_x_axis, const std::string & out_y_axis, const std::string & out_x_axis_unit, const std::string & out_y_axis_unit) throw (CDMException) [virtual]`

@ brief change the (main) projection of the dataReaders cdm to this new projection

Parameters:

method Interpolation method
proj_input input-string for proj4, used as output projection
out_x_axis config-string for x_axis, either '1,2,...,5' or 'auto' or 'auto,distance=3.5'
out_y_axis config-string for y_axis, either '1,2,...,5' or 'auto' or 'auto,distance=3.5'
out_x_axis_unit unit of the output x-axis
out_y_axis_unit unit of the output y-axis

14.14.2.4 `virtual void MetNoFimex::CDMInterpolator::setLatitudeName (const std::string & latName) [inline, virtual]`

set the name for the automatically generated latitude coordinate axis. This must be set before changeProjection is called.

Parameters:

latName name for latitude

14.14.2.5 `virtual const std::string& MetNoFimex::CDMInterpolator::getLatitudeName () const [inline, virtual]`

Returns:

the name used for latitude in the automatic coordinate generation

14.14.2.6 virtual void MetNoFimex::CDMInterpolator::setLongitudeName (const std::string & *lonName*) [inline, virtual]

set the name for the automatically generated longitude coordinate axis. This must be set before changeProjection is called.

Parameters:

latName name for longitude

14.14.2.7 virtual const std::string& MetNoFimex::CDMInterpolator::getLongitudeName () const [inline, virtual]**Returns:**

the name used for longitude in the automatic coordinate generation

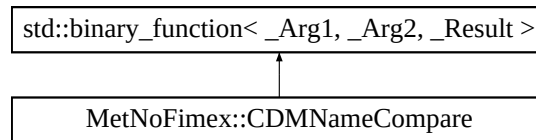
The documentation for this class was generated from the following file:

- include/fimex/[CDMInterpolator.h](#)

14.15 MetNoFimex::CDMNameCompare Struct Reference

```
#include <CDMNamedEntity.h>
```

Inheritance diagram for MetNoFimex::CDMNameCompare::



Public Member Functions

- `int operator()` (const [CDMNamedEntity](#) &*e1*, const [CDMNamedEntity](#) &*e2*)

14.15.1 Detailed Description

functor to compares names of two [CDMNamedEntity](#) using `std::string::compare`

14.15.2 Member Function Documentation

14.15.2.1 `int MetNoFimex::CDMNameCompare::operator()` (const [CDMNamedEntity](#) & *e1*, const [CDMNamedEntity](#) & *e2*) [`inline`]

References `std::basic_string<_CharT, _Traits, _Alloc>::compare()`, and `MetNoFimex::CDMNamedEntity::getName()`.

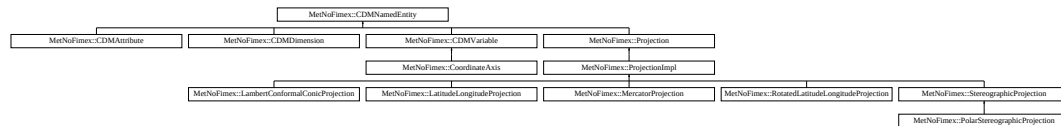
The documentation for this struct was generated from the following file:

- `include/fimex/CDMNamedEntity.h`

14.16 MetNoFimex::CDMNamedEntity Class Reference

```
#include <CDMNamedEntity.h>
```

Inheritance diagram for MetNoFimex::CDMNamedEntity::



Public Member Functions

- virtual [~CDMNamedEntity](#) ()=0
- virtual const **std::string** & [getName](#) () const =0

14.16.1 Detailed Description

interface for all [CDM](#) Entities (variable, attribute, dimension) which support the 'getName' method

14.16.2 Constructor & Destructor Documentation

14.16.2.1 virtual MetNoFimex::CDMNamedEntity::~~CDMNamedEntity () [pure virtual]

14.16.3 Member Function Documentation

14.16.3.1 virtual const **std::string**& MetNoFimex::CDMNamedEntity::getName () const [pure virtual]

Implemented in [MetNoFimex::CDMAttribute](#), [MetNoFimex::CDMDimension](#), [MetNoFimex::CDMVariable](#), [MetNoFimex::Projection](#), and [MetNoFimex::ProjectionImpl](#).

Referenced by [MetNoFimex::CDMNameEqual::operator\(\)](#), and [MetNoFimex::CDMNameCompare::operator\(\)](#).

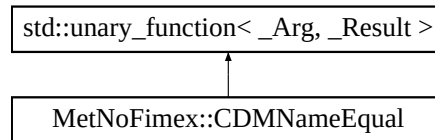
The documentation for this class was generated from the following file:

- include/fimex/[CDMNamedEntity.h](#)

14.17 MetNoFimex::CDMNameEqual Class Reference

```
#include <CDMNamedEntity.h>
```

Inheritance diagram for MetNoFimex::CDMNameEqual::



Public Member Functions

- [CDMNameEqual](#) (**std::string** name)
- [CDMNameEqual](#) (const [CDMNamedEntity](#) &entity)
- [~CDMNameEqual](#) ()
- bool [operator\(\)](#) (const [CDMNamedEntity](#) &e)

14.17.1 Detailed Description

functor to find a [CDMNamedEntity](#) equal to the set name using `std::string::operator==`

14.17.2 Constructor & Destructor Documentation

14.17.2.1 `MetNoFimex::CDMNameEqual::CDMNameEqual (std::string name)` [`inline`, `explicit`]

14.17.2.2 `MetNoFimex::CDMNameEqual::CDMNameEqual (const CDMNamedEntity & entity)` [`inline`, `explicit`]

14.17.2.3 `MetNoFimex::CDMNameEqual::~~CDMNameEqual ()` [`inline`]

14.17.3 Member Function Documentation

14.17.3.1 `bool MetNoFimex::CDMNameEqual::operator() (const CDMNamedEntity & e)` [`inline`]

References `MetNoFimex::CDMNamedEntity::getName()`.

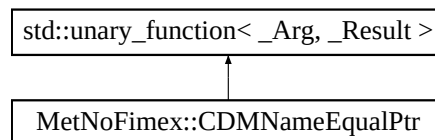
The documentation for this class was generated from the following file:

- `include/fimex/CDMNamedEntity.h`

14.18 MetNoFimex::CDMNameEqualPtr Class Reference

```
#include <CDMNamedEntity.h>
```

Inheritance diagram for MetNoFimex::CDMNameEqualPtr::



Public Member Functions

- [CDMNameEqualPtr](#) (**std::string** name)
- [CDMNameEqualPtr](#) (const boost::shared_ptr< const [CDMNamedEntity](#) > &entity)
- [~CDMNameEqualPtr](#) ()
- bool [operator\(\)](#) (const boost::shared_ptr< const [CDMNamedEntity](#) > &e)

14.18.1 Detailed Description

functor to find a boost::shared_ptr<CDMNamedEntity> equal to the set name using std::string::operator==

14.18.2 Constructor & Destructor Documentation

14.18.2.1 [MetNoFimex::CDMNameEqualPtr::CDMNameEqualPtr](#) (**std::string** name)
 [inline, explicit]

14.18.2.2 [MetNoFimex::CDMNameEqualPtr::CDMNameEqualPtr](#) (const boost::shared_ptr< const [CDMNamedEntity](#) > &entity) [inline, explicit]

14.18.2.3 [MetNoFimex::CDMNameEqualPtr::~~CDMNameEqualPtr](#) () [inline]

14.18.3 Member Function Documentation

14.18.3.1 bool [MetNoFimex::CDMNameEqualPtr::operator\(\)](#) (const boost::shared_ptr< const [CDMNamedEntity](#) > &e) [inline]

The documentation for this class was generated from the following file:

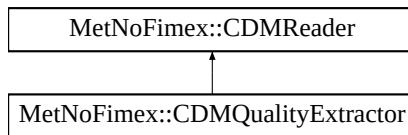
- include/fimex/[CDMNamedEntity.h](#)

14.19 MetNoFimex::CDMQualityExtractor Class Reference

Extract data with defined quality status.

```
#include <CDMQualityExtractor.h>
```

Inheritance diagram for MetNoFimex::CDMQualityExtractor::



Public Member Functions

- [CDMQualityExtractor](#) (boost::shared_ptr< [CDMReader](#) > dataReader, **std::string** autoConfString="", **std::string** configFile="") throw (CDMException)
- virtual [~CDMQualityExtractor](#) ()
- virtual boost::shared_ptr< [Data](#) > [getDataSlice](#) (const **std::string** &varName, size_t unLimDimPos=0) throw (CDMException)
- const **std::map**< **std::string**, **std::string** > [getStatusVariable](#) () const
- const **std::map**< **std::string**, **std::string** > [getVariableFlags](#) () const
- const **std::map**< **std::string**, **std::vector**< double > > [getVariableValues](#) () const

14.19.1 Detailed Description

Extract data with defined quality status.

The [CDMQualityExtractor](#) will select data from data-sources matching only configurable quality constraints. [Data](#) not matching these constraints will be set to undefined.

The configuration works either semi-automatic by interpreting the quality flags as given in CF-1.x at <http://cf-pcmdi.llnl.gov/documents/cf-conventions/1.4/cf-conventions.html#flags> or by using a configuration-file describing the quality-relations between the different variables.

All variables with no quality-configuration will not be changed.

Warning:

The [CDMQualityExtractor](#) will read the status-variable after applying eventual quality-flags to them. It is therefore the task of the writer of the configuration, that no circular quality-flags exist.

14.19.2 Constructor & Destructor Documentation

14.19.2.1 MetNoFimex::CDMQualityExtractor::CDMQualityExtractor (boost::shared_ptr< [CDMReader](#) > dataReader, **std::string** autoConfString = "", **std::string** configFile = "") throw (CDMException)

Initialize the [CDMQualityExtractor](#)

Parameters:

dataReader the data-source

autoConfString the default value for CF-1.4 compatible status_flags, i.e. "all, highest, lowest, values=0,1,...,3", the values here might be overwritten by the config-file. If empty, no quality extraction on the basis of CF-1.4 will be used.

configFile filename of a cdmQualityConfig.xml file. If empty, no quality-file will be used.

14.19.2.2 `virtual MetNoFimex::CDMQualityExtractor::~~CDMQualityExtractor () [inline, virtual]`

14.19.3 Member Function Documentation

14.19.3.1 `virtual boost::shared_ptr<Data> MetNoFimex::CDMQualityExtractor::getDataSlice (const std::string & varName, size_t unLimDimPos = 0) throw (CDMException) [virtual]`

Read and manipulate the data

14.19.3.2 `const std::map<std::string, std::string> MetNoFimex::CDMQualityExtractor::getStatusVariable () const [inline]`

Read the internals of statusVariable. This code is mainly thought for testing/debugging.

14.19.3.3 `const std::map<std::string, std::string> MetNoFimex::CDMQualityExtractor::getVariableFlags () const [inline]`

Read the internals of variableFlags, for testing/debugging.

14.19.3.4 `const std::map<std::string, std::vector<double> > MetNoFimex::CDMQualityExtractor::getVariableValues () const [inline]`

Read the internals of variableValues, for testing/debugging.

The documentation for this class was generated from the following file:

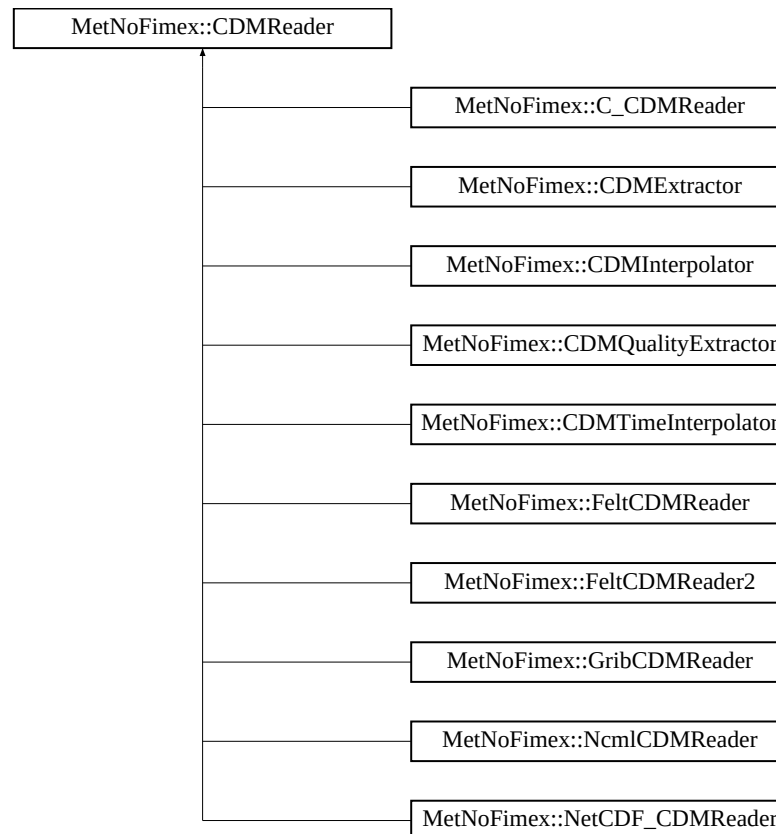
- `include/fimex/CDMQualityExtractor.h`

14.20 MetNoFimex::CDMReader Class Reference

Basic interface for [CDM](#) reading and manipulation classes.

```
#include <CDMReader.h>
```

Inheritance diagram for MetNoFimex::CDMReader::



Public Member Functions

- [CDMReader](#) ()
- virtual [~CDMReader](#) ()
- virtual const [CDM](#) & [getCDM](#) () const
- virtual boost::shared_ptr< [Data](#) > [getDataSlice](#) (const **std::string** &varName, size_t unLimDim-Pos)=0 throw (CDMException)
data-reading function to be called from the [CDMWriter](#)
- virtual boost::shared_ptr< [Data](#) > [getDataSlice](#) (const **std::string** &varName, const [SliceBuilder](#) &sb) throw (CDMException)
data-reading function to be called from the [CDMWriter](#)
- virtual boost::shared_ptr< [Data](#) > [getData](#) (const **std::string** &varName) throw (CDMException)
data-reading function to be called from the [CDMWriter](#)

- virtual boost::shared_ptr< [Data](#) > [getScaledDataSlice](#) (const **std::string** &varName, size_t unLimDimPos) throw (CDMException)
read and scale a dataslice
- virtual boost::shared_ptr< [Data](#) > [getScaledDataSliceInUnit](#) (const **std::string** &varName, const **std::string** &unit, size_t unLimDimPos) throw (CDMException)
read and scale a dataslice to a known unit
- virtual boost::shared_ptr< [Data](#) > [getScaledDataSlice](#) (const **std::string** &varName, const [SliceBuilder](#) &sb) throw (CDMException)
read and scale a dataslice
- virtual boost::shared_ptr< [Data](#) > [getScaledDataSliceInUnit](#) (const **std::string** &varName, const **std::string** &unit, const [SliceBuilder](#) &sb) throw (CDMException)
read and scale a dataslice to a set unit
- virtual boost::shared_ptr< [Data](#) > [getScaledData](#) (const **std::string** &varName) throw (CDMException)
read and scale the complete data
- virtual boost::shared_ptr< [Data](#) > [getScaledDataInUnit](#) (const **std::string** &varName, const **std::string** &unit) throw (CDMException)
read and scale the complete data to a set unit

Protected Member Functions

- virtual boost::shared_ptr< [Data](#) > [getDataSliceFromMemory](#) (const [CDMVariable](#) &variable, size_t unLimDimPos=0) throw (CDMException)

Protected Attributes

- boost::shared_ptr< [CDM](#) > [cdm_](#)

14.20.1 Detailed Description

Basic interface for [CDM](#) reading and manipulation classes.

The [CDMReader](#) is the basic interface for reading and manipulation of the cdm datastructure. The [CDMWriter](#) will work with an implementation of the [CDMReader](#) and read the included data in the cdm or the data provided through the implementation of the [CDMReader#getDataSlice](#)

See also:

[FeltCDMReader](#)

14.20.2 Constructor & Destructor Documentation

14.20.2.1 `MetNoFimex::CDMReader::CDMReader ()`

14.20.2.2 `virtual MetNoFimex::CDMReader::~~CDMReader ()` `[inline, virtual]`

14.20.3 Member Function Documentation

14.20.3.1 `virtual const CDM& MetNoFimex::CDMReader::getCDM () const` `[virtual]`

Retrieve the cdm structure of this reader.

14.20.3.2 `virtual boost::shared_ptr<Data> MetNoFimex::CDMReader::getDataSlice (const std::string & varName, size_t unLimDimPos) throw (CDMException)` `[pure virtual]`

data-reading function to be called from the [CDMWriter](#)

This methods needs to be implemented by the [CDMReader](#). It should provide the data for each variable, either by reading from disk, converting from another [CDMReader](#) or reading from an in-memory data-section.

This function should retrieve the whole data for a dataset without unlimited dimension if the `unLimDimPos == 0`.

Parameters:

varName name of the variable to read

unLimDimPos (optional) if the variable contains a unlimited dimension (max one allowed) an slice of this position is returned

Implemented in [MetNoFimex::C_CDMReader](#), [MetNoFimex::CDMExtractor](#), [MetNoFimex::CDMInterpolator](#), [MetNoFimex::FeltCDMReader2](#), [MetNoFimex::GribCDMReader](#), [MetNoFimex::NcmlCDMReader](#), and [MetNoFimex::NetCDF_CDMReader](#).

14.20.3.3 `virtual boost::shared_ptr<Data> MetNoFimex::CDMReader::getDataSlice (const std::string & varName, const SliceBuilder & sb) throw (CDMException)` `[virtual]`

data-reading function to be called from the [CDMWriter](#)

Parameters:

varName name of the variable to read

sb a [SliceBuilder](#) generated from this CDMReaders [CDM](#) This method has a default implementation depending on `getDataSlice(varName, unLimDimPos)`, but should be implemented for performance reasons.

Reimplemented in [MetNoFimex::NetCDF_CDMReader](#).

14.20.3.4 `virtual boost::shared_ptr<Data> MetNoFimex::CDMReader::getData (const std::string & varName) throw (CDMException)` `[virtual]`

data-reading function to be called from the [CDMWriter](#)

The `getData` function is a convenient function to retrieve all data from a file. It is implemented using `getDataSlice`. It should be used with care, since a complete variable might be bigger than available memory.

Parameters:

varName name of the variable to read

14.20.3.5 `virtual boost::shared_ptr<Data> MetNoFimex::CDMReader::getScaledDataSlice (const std::string & varName, size_t unLimDimPos) throw (CDMException)`
[virtual]

read and scale a dataslice

This functions uses `getDataSlice` internally. It tries to read "scale_factor" "add_offset" and "_FillValue" and apply the scaling to the read data. Output-datatype will be double, output `_FillValue` will be `MIFI_UNDEFINED_D`

Parameters:

varName name of the variable to read

unLimDimPos (optional) if the variable contains a unlimited dimension (max one allowed) an slice of this position is returned

14.20.3.6 `virtual boost::shared_ptr<Data> MetNoFimex::CDMReader::getScaledDataSliceInUnit (const std::string & varName, const std::string & unit, size_t unLimDimPos) throw (CDMException)`
[virtual]

read and scale a dataslice to a known unit

This functions uses `getDataSlice` internally. It tries to read "scale_factor" "add_offset" and "_FillValue" and apply the scaling to the read data. Output-datatype will be double, output `_FillValue` will be `MIFI_UNDEFINED_D`. The data will be converted to match unit.

Parameters:

varName name of the variable to read

unit unit-string

unLimDimPos (optional) if the variable contains a unlimited dimension (max one allowed) an slice of this position is returned

14.20.3.7 `virtual boost::shared_ptr<Data> MetNoFimex::CDMReader::getScaledDataSlice (const std::string & varName, const SliceBuilder & sb) throw (CDMException)`
[virtual]

read and scale a dataslice

Parameters:

varName name of the variable to read

sb [SliceBuilder](#) to restrict the data

See also:

`getScaledDataSlice(varName, unLimDimPos)`

14.20.3.8 `virtual boost::shared_ptr<Data> MetNoFimex::CDMReader::getScaledDataSliceInUnit (const std::string & varName, const std::string & unit, const SliceBuilder & sb) throw (CDMException)` [virtual]

read and scale a dataslice to a set unit

Parameters:

varName name of the variable to read

unit unit string to scale to

sb [SliceBuilder](#) to restrict the data

See also:

`getScaledDataSlice(varName, unLimDimPos)`

14.20.3.9 `virtual boost::shared_ptr<Data> MetNoFimex::CDMReader::getScaledData (const std::string & varName) throw (CDMException)` [virtual]

read and scale the complete data

This functions uses `getData` internally. It tries to read "scale_factor" "add_offset" and "_FillValue" and apply the scaling to the read data. Output-datatype will be double, output _FillValue will be MIFI_UNDEFINED_D.

Parameters:

varName name of the variable to read

14.20.3.10 `virtual boost::shared_ptr<Data> MetNoFimex::CDMReader::getScaledDataInUnit (const std::string & varName, const std::string & unit) throw (CDMException)` [virtual]

read and scale the complete data to a set unit

This functions uses `getData` internally. It tries to read "scale_factor" "add_offset" and "_FillValue" and apply the scaling to the read data. Output-datatype will be double, output _FillValue will be MIFI_UNDEFINED_D. The data will be converted to match unit.

Parameters:

varName name of the variable to read

unit the unit-string to convert the data to

14.20.3.11 `virtual boost::shared_ptr<Data> MetNoFimex::CDMReader::getDataSliceFromMemory (const CDMVariable & variable, size_t unLimDimPos = 0) throw (CDMException)` [protected, virtual]

Read the data from the variable.hasData() and select the correct unLimDimPos. This function should be used internally from getDataSlice.

Parameters:

variable the variable to read data from

unLimDimPos (optional) the unlimited position

14.20.4 Member Data Documentation

14.20.4.1 `boost::shared_ptr<CDM> MetNoFimex::CDMReader::cdm_` [protected]

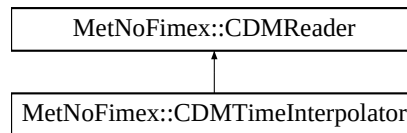
The documentation for this class was generated from the following file:

- include/fimex/[CDMReader.h](#)

14.21 MetNoFimex::CDMTimeInterpolator Class Reference

```
#include <CDMTimeInterpolator.h>
```

Inheritance diagram for MetNoFimex::CDMTimeInterpolator::



Public Member Functions

- [CDMTimeInterpolator](#) (boost::shared_ptr< [CDMReader](#) > dataReader)
- virtual [~CDMTimeInterpolator](#) ()
- virtual boost::shared_ptr< [Data](#) > [getDataSlice](#) (const **std::string** &varName, size_t unLimDimPos=0) throw (CDMException)
retrieve data from the underlying dataReader and interpolate the values due to the current projection
- virtual void [changeTimeAxis](#) (**std::string** timeSpec) throw (CDMException)

14.21.1 Constructor & Destructor Documentation

14.21.1.1 MetNoFimex::CDMTimeInterpolator::CDMTimeInterpolator (boost::shared_ptr< CDMReader > dataReader)

14.21.1.2 virtual MetNoFimex::CDMTimeInterpolator::~~CDMTimeInterpolator () [virtual]

14.21.2 Member Function Documentation

14.21.2.1 virtual boost::shared_ptr<Data> MetNoFimex::CDMTimeInterpolator::getDataSlice (const **std::string** & varName, size_t unLimDimPos = 0) throw (CDMException)
 [virtual]

retrieve data from the underlying dataReader and interpolate the values due to the current projection

Parameters:

varName name of variable

size_t unLimDimPos position of the unlimited dimension, most commonly time-position of the output as set in [changeTimeAxis](#)

14.21.2.2 virtual void MetNoFimex::CDMTimeInterpolator::changeTimeAxis (**std::string** timeSpec) throw (CDMException) [virtual]

change the time-axis from from the one given to a new specification

Parameters:

timeSpec string of time-specification

Exceptions:

[*CDMException*](#) on unparsable timeSpec

See also:

secTimeSpec

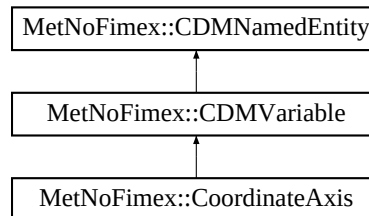
The documentation for this class was generated from the following file:

- include/fimex/[CDMTimeInterpolator.h](#)

14.22 MetNoFimex::CDMVariable Class Reference

```
#include <fimex/CDMVariable.h>
```

Inheritance diagram for MetNoFimex::CDMVariable::



Public Member Functions

- [CDMVariable](#) (**std::string** name, [CDMDataType](#) datatype, **std::vector**< **std::string** > shape)
- virtual [~CDMVariable](#) ()
- const **std::string** & [getName](#) () const
- void [setName](#) (**std::string** newName)
- [CDMDataType](#) [getDataType](#) () const
- void [setDataType](#) ([CDMDataType](#) type)
- const **std::vector**< **std::string** > & [getShape](#) () const
- void [setShape](#) (**std::vector**< **std::string** > newShape)
- void [setAsSpatialVector](#) (const **std::string** &counterpart, const **std::string** &direction)
- bool [isSpatialVector](#) () const
- const **std::string** & [getSpatialVectorCounterpart](#) () const
get the spatial counterpart of this vector
- const **std::string** & [getSpatialVectorDirection](#) () const
get the possible directions of this spatial vector (comma-separated string)
- bool [checkDimension](#) (const **std::string** &dimension) const
- void [toXMLStream](#) (**std::ostream** &out) const
print a xml representation to the stream without attributes
- void [toXMLStream](#) (**std::ostream** &out, const **std::vector**< [CDMAttribute](#) > &attrs) const
print a xml representation to the stream with attributes
- void [setData](#) (boost::shared_ptr< [Data](#) > data)
add data to the variable
- const boost::shared_ptr< [Data](#) > [getData](#) () const
retrieve volatile data from this variable
- int [hasData](#) () const
check if real data has been set with [setData\(\)](#) (null-pointer reference returns false)

14.22.1 Constructor & Destructor Documentation

14.22.1.1 `MetNoFimex::CDMVariable::CDMVariable (std::string name, CDMDataType datatype, std::vector< std::string > shape)` `[explicit]`

14.22.1.2 `virtual MetNoFimex::CDMVariable::~~CDMVariable ()` `[virtual]`

14.22.2 Member Function Documentation

14.22.2.1 `const std::string& MetNoFimex::CDMVariable::getName () const` `[inline, virtual]`

Implements [MetNoFimex::CDMNamedEntity](#).

Referenced by `MetNoFimex::CoordinateAxis::operator<()`.

14.22.2.2 `void MetNoFimex::CDMVariable::setName (std::string newName)` `[inline]`

14.22.2.3 `CDMDataType MetNoFimex::CDMVariable::getDataType () const` `[inline]`

14.22.2.4 `void MetNoFimex::CDMVariable::setDataType (CDMDataType type)` `[inline]`

14.22.2.5 `const std::vector<std::string>& MetNoFimex::CDMVariable::getShape () const` `[inline]`

14.22.2.6 `void MetNoFimex::CDMVariable::setShape (std::vector< std::string > newShape)` `[inline]`

14.22.2.7 `void MetNoFimex::CDMVariable::setAsSpatialVector (const std::string & counterpart, const std::string & direction)`

Declare this variable to be part of a spatial vector, e.g. (x-wind, y-wind)

Parameters:

counterpart name of the other variable being part of this vector

direction comma-separated list of possible directions for this vector, e.g. "x,longitude"

14.22.2.8 `bool MetNoFimex::CDMVariable::isSpatialVector () const` `[inline]`

test if this variable has been declared to be a spatial vector

14.22.2.9 `const std::string& MetNoFimex::CDMVariable::getSpatialVectorCounterpart () const` `[inline]`

get the spatial counterpart of this vector

14.22.2.10 `const std::string& MetNoFimex::CDMVariable::getSpatialVectorDirection () const` `[inline]`

get the possible directions of this spatial vector (comma-separated string)

14.22.2.11 **bool MetNoFimex::CDMVariable::checkDimension (const std::string & *dimension*) const**

check the dimension of a variable

Parameters:

dimension the dimension to check for

14.22.2.12 **void MetNoFimex::CDMVariable::toXMLStream (std::ostream & *out*) const**

print a xml representation to the stream without attributes

14.22.2.13 **void MetNoFimex::CDMVariable::toXMLStream (std::ostream & *out*, const std::vector< CDMAttribute > & *attrs*) const**

print a xml representation to the stream with attributes

14.22.2.14 **void MetNoFimex::CDMVariable::setData (boost::shared_ptr< Data > *data*) [inline]**

add data to the variable

14.22.2.15 **const boost::shared_ptr<Data> MetNoFimex::CDMVariable::getData () const [inline]**

retrieve volatile data from this variable

Retrieve data, but only if it has been set previously by [setData\(\)](#) this method will not try to read data from the disk. Use [CDMReader::getData\(const std::string& varName\)](#) to get the data from memory or from disk.

14.22.2.16 **int MetNoFimex::CDMVariable::hasData () const [inline]**

check if real data has been set with [setData\(\)](#) (null-pointer reference returns false)

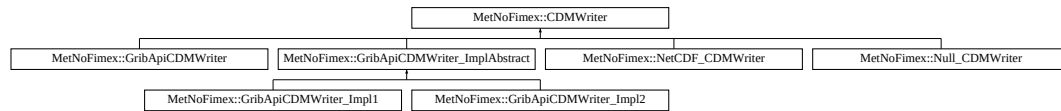
The documentation for this class was generated from the following file:

- include/fimex/[CDMVariable.h](#)

14.23 MetNoFimex::CDMWriter Class Reference

```
#include <CDMWriter.h>
```

Inheritance diagram for MetNoFimex::CDMWriter::



Public Member Functions

- [CDMWriter](#) (boost::shared_ptr< [CDMReader](#) > [cdmReader](#), const std::string &[outputFile](#))
- virtual [~CDMWriter](#) ()

Protected Attributes

- boost::shared_ptr< [CDMReader](#) > [cdmReader](#)
- const std::string [outputFile](#)

14.23.1 Constructor & Destructor Documentation

14.23.1.1 [MetNoFimex::CDMWriter::CDMWriter](#) (boost::shared_ptr< [CDMReader](#) > [cdmReader](#), const std::string & [outputFile](#)) `[inline]`

14.23.1.2 [virtual MetNoFimex::CDMWriter::~~CDMWriter](#) () `[inline, virtual]`

14.23.2 Member Data Documentation

14.23.2.1 [boost::shared_ptr<CDMReader> MetNoFimex::CDMWriter::cdmReader](#) `[protected]`

14.23.2.2 [const std::string MetNoFimex::CDMWriter::outputFile](#) `[protected]`

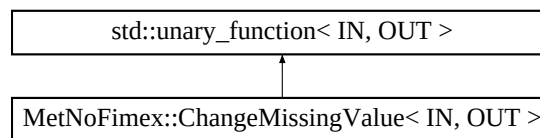
The documentation for this class was generated from the following file:

- [include/fimex/CDMWriter.h](#)

14.24 MetNoFimex::ChangeMissingValue< IN, OUT > Class Template Reference

```
#include <Utils.h>
```

Inheritance diagram for MetNoFimex::ChangeMissingValue< IN, OUT >::



Public Member Functions

- [ChangeMissingValue](#) (double oldFill, double newFill)
- OUT [operator\(\)](#) (const IN &in) const

14.24.1 Detailed Description

```
template<typename IN, typename OUT> class MetNoFimex::ChangeMissingValue< IN, OUT >
```

Change the missing value

14.24.2 Constructor & Destructor Documentation

14.24.2.1 `template<typename IN, typename OUT> MetNoFimex::ChangeMissingValue< IN, OUT >::ChangeMissingValue (double oldFill, double newFill) [inline]`

14.24.3 Member Function Documentation

14.24.3.1 `template<typename IN, typename OUT> OUT MetNoFimex::ChangeMissingValue< IN, OUT >::operator() (const IN &in) const [inline]`

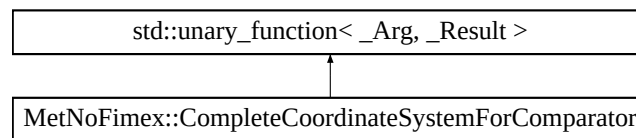
The documentation for this class was generated from the following file:

- include/fimex/[Utils.h](#)

14.25 MetNoFimex::CompleteCoordinateSystemForComparator Struct Reference

```
#include <CoordinateSystem.h>
```

Inheritance diagram for MetNoFimex::CompleteCoordinateSystemForComparator::



Public Member Functions

- [CompleteCoordinateSystemForComparator](#) (const **std::string** &varName)
- virtual [~CompleteCoordinateSystemForComparator](#) ()
- bool [operator\(\)](#) (const boost::shared_ptr< const [CoordinateSystem](#) > &cs)

14.25.1 Detailed Description

Functor to check if a coordinate system completely describes a variable, i.e. all axes match fully.

14.25.2 Constructor & Destructor Documentation

14.25.2.1 `MetNoFimex::CompleteCoordinateSystemForComparator::CompleteCoordinateSystemForComparator (const std::string & varName) [inline]`

14.25.2.2 `virtual MetNoFimex::CompleteCoordinateSystemForComparator::~~CompleteCoordinateSystemForComparator () [inline, virtual]`

14.25.3 Member Function Documentation

14.25.3.1 `bool MetNoFimex::CompleteCoordinateSystemForComparator::operator() (const boost::shared_ptr< const CoordinateSystem > & cs) [inline]`

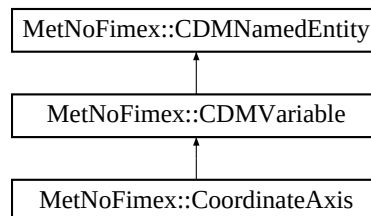
The documentation for this struct was generated from the following file:

- include/fimex/coordSys/[CoordinateSystem.h](#)

14.26 MetNoFimex::CoordinateAxis Class Reference

```
#include <fimex/coordSys/CoordinateAxis.h>
```

Inheritance diagram for MetNoFimex::CoordinateAxis::



Public Types

- enum [AxisType](#) {
[Undefined](#) = 0, [GeoX](#), [GeoY](#), [GeoZ](#),
[Time](#), [Lon](#), [Lat](#), [Pressure](#),
[Height](#) }

Public Member Functions

- [CoordinateAxis](#) (const [CDMVariable](#) &var)
- virtual [~CoordinateAxis](#) ()
- bool [operator<](#) (const [CoordinateAxis](#) &ca)
- [AxisType](#) [getAxisType](#) () const
- [std::string](#) [getAxisTypeStr](#) () const
- void [setAxisType](#) ([AxisType](#) t)
- bool [isAxisType](#) ([AxisType](#) t) const
- bool [isExplicit](#) () const
- void [setExplicit](#) (bool isExplicit)

Static Public Member Functions

- static [std::string](#) [type2string](#) ([AxisType](#) type)

14.26.1 Member Enumeration Documentation

14.26.1.1 enum MetNoFimex::CoordinateAxis::AxisType

Enumerator:

Undefined
GeoX
GeoY
GeoZ

Time
Lon
Lat
Pressure
Height

14.26.2 Constructor & Destructor Documentation

14.26.2.1 `MetNoFimex::CoordinateAxis::CoordinateAxis (const CDMVariable & var)`
`[inline, explicit]`

14.26.2.2 `virtual MetNoFimex::CoordinateAxis::~~CoordinateAxis ()` `[inline, virtual]`

14.26.3 Member Function Documentation

14.26.3.1 `static std::string MetNoFimex::CoordinateAxis::type2string (AxisType type)`
`[inline, static]`

References GeoX, GeoY, GeoZ, Height, Lat, Lon, Pressure, and Time.

Referenced by getAxisTypeStr().

14.26.3.2 `bool MetNoFimex::CoordinateAxis::operator< (const CoordinateAxis & ca)`
`[inline]`

References MetNoFimex::CDMVariable::getName().

14.26.3.3 `AxisType MetNoFimex::CoordinateAxis::getAxisType () const` `[inline]`

14.26.3.4 `std::string MetNoFimex::CoordinateAxis::getAxisTypeStr () const` `[inline]`

References type2string().

14.26.3.5 `void MetNoFimex::CoordinateAxis::setAxisType (AxisType t)` `[inline]`

14.26.3.6 `bool MetNoFimex::CoordinateAxis::isAxisType (AxisType t) const` `[inline]`

14.26.3.7 `bool MetNoFimex::CoordinateAxis::isExplicit () const` `[inline]`

Check if this axis is a explicitly netcdf-dimension, too.

Returns:

false if this is a implicit 'coordinates' variable, rather than a netcdf-dimension

14.26.3.8 `void MetNoFimex::CoordinateAxis::setExplicit (bool isExplicit)` `[inline]`

The documentation for this class was generated from the following file:

- include/fimex/coordSys/[CoordinateAxis.h](#)

14.27 MetNoFimex::CoordinateSystem Class Reference

```
#include <fimex/coordSys/CoordinateSystem.h>
```

Public Types

- typedef boost::shared_ptr< const [CoordinateAxis](#) > [ConstAxisPtr](#)
- typedef boost::shared_ptr< [CoordinateAxis](#) > [AxisPtr](#)
- typedef **std::vector**< [ConstAxisPtr](#) > [ConstAxisList](#)

Public Member Functions

- [CoordinateSystem](#) ()
- [CoordinateSystem](#) (const **std::string** &conventionName)
- virtual [~CoordinateSystem](#) ()
- virtual **std::string** [id](#) () const
- virtual **std::string** [getConventionName](#) () const
- virtual void [setConventionName](#) (const **std::string** &conventionName)
- virtual bool [isComplete](#) (const **std::string** &varName) const
- virtual void [setComplete](#) (const **std::string** &varName, bool set=true)
- virtual bool [isCSFor](#) (const **std::string** &varName) const
- virtual void [setCSFor](#) (const **std::string** &varName, bool set=true)
- virtual bool [isSimpleSpatialGridded](#) () const
- virtual void [setSimpleSpatialGridded](#) (bool set=true)
- virtual bool [hasProjection](#) () const
- virtual boost::shared_ptr< const [Projection](#) > [getProjection](#) () const
- virtual void [setProjection](#) (boost::shared_ptr< const [Projection](#) > proj)
- virtual bool [hasAxisType](#) ([CoordinateAxis::AxisType](#) type) const
- virtual [ConstAxisPtr](#) [findAxisOfType](#) ([CoordinateAxis::AxisType](#) type) const
- virtual [ConstAxisPtr](#) [findAxisOfType](#) (const **std::vector**< [CoordinateAxis::AxisType](#) > &types) const
- virtual [ConstAxisPtr](#) [getGeoXAxis](#) () const
- virtual [ConstAxisPtr](#) [getGeoYAxis](#) () const
- virtual [ConstAxisPtr](#) [getGeoZAxis](#) () const
- virtual [ConstAxisPtr](#) [getTimeAxis](#) () const
- virtual **ConstAxisList** [getAxes](#) () const
- virtual void [setAxis](#) ([AxisPtr](#) axis)

14.27.1 Detailed Description

CoordinateSystems are usually created using the [listCoordinateSystems\(const CDM& cdm\)](#) function, see example there.

To investigate the coordinate systems of a file, use {*fimex* -printCS}.

14.27.2 Member Typedef Documentation

14.27.2.1 typedef boost::shared_ptr<const [CoordinateAxis](#)> [MetNoFimex::CoordinateSystem::ConstAxisPtr](#)

a garbage collected pointer to a constant coordinateAxis

14.27.2.2 `typedef boost::shared_ptr<CoordinateAxis> MetNoFimex::CoordinateSystem::AxisPtr`

a garbage collected pointer to a [CoordinateAxis](#)

14.27.2.3 `typedef std::vector<ConstAxisPtr> MetNoFimex::CoordinateSystem::ConstAxisList`

a list to constant axis pointer

14.27.3 Constructor & Destructor Documentation

14.27.3.1 `MetNoFimex::CoordinateSystem::CoordinateSystem ()`

CoordinateSystems are usually created within the [listCoordinateSystems\(const CDM& cdm\)](#) funcion.

14.27.3.2 `MetNoFimex::CoordinateSystem::CoordinateSystem (const std::string & conventionName) [explicit]`

14.27.3.3 `virtual MetNoFimex::CoordinateSystem::~~CoordinateSystem () [inline, virtual]`

14.27.4 Member Function Documentation

14.27.4.1 `virtual std::string MetNoFimex::CoordinateSystem::id () const [virtual]`

unique identifier for a coordinate system

14.27.4.2 `virtual std::string MetNoFimex::CoordinateSystem::getConventionName () const [virtual]`

Returns:

the name of convention used to build the CS

14.27.4.3 `virtual void MetNoFimex::CoordinateSystem::setConventionName (const std::string & conventionName) [virtual]`

set the convention name

Parameters:

conventionName

14.27.4.4 `virtual bool MetNoFimex::CoordinateSystem::isComplete (const std::string & varName) const [virtual]`

All axes of this system are used by the variable varName

Parameters:

varName variable name

14.27.4.5 `virtual void MetNoFimex::CoordinateSystem::setComplete (const std::string & varName, bool set = true) [virtual]`

set or unset if the coordinate system is complete for the variable

14.27.4.6 `virtual bool MetNoFimex::CoordinateSystem::isCSFor (const std::string & varName) const [virtual]`

All dimensions of the variable are described by [CoordinateSystem](#)

Parameters:

varName variable name

14.27.4.7 `virtual void MetNoFimex::CoordinateSystem::setCSFor (const std::string & varName, bool set = true) [virtual]`

Set or unset if all dimensions are described by the [CoordinateSystem](#)

14.27.4.8 `virtual bool MetNoFimex::CoordinateSystem::isSimpleSpatialGridded () const [virtual]`

Check if coordinate system has direct spatial axes, i.e. 1-dim x,y axes or 1-dim lon,lat axes

14.27.4.9 `virtual void MetNoFimex::CoordinateSystem::setSimpleSpatialGridded (bool set = true) [virtual]`

Set or unset if this coordinate system has spatial axes, i.e. 1-dim x,y or lon/lat axes

14.27.4.10 `virtual bool MetNoFimex::CoordinateSystem::hasProjection () const [virtual]`

Check if the coordinate-system has a projection (of GeoX, GeoY and optionally GeoZ)

14.27.4.11 `virtual boost::shared_ptr<const Projection> MetNoFimex::CoordinateSystem::getProjection () const [virtual]`

Get the projection of the coordinate-system (projection of GeoX, GeoY and optionally GeoZ)

Returns:

projection, or null ptr

14.27.4.12 `virtual void MetNoFimex::CoordinateSystem::setProjection (boost::shared_ptr< const Projection > proj) [virtual]`

Set the projection of the coordinate-system (projection of GeoX, GeoY and optionally GeoZ)

14.27.4.13 virtual bool MetNoFimex::CoordinateSystem::hasAxisType (CoordinateAxis::AxisType *type*) const [virtual]

Check if the [CoordinateSystem](#) contains exactly the axis type

Parameters:

type axis type to check against

14.27.4.14 virtual ConstAxisPtr MetNoFimex::CoordinateSystem::findAxisOfType (CoordinateAxis::AxisType *type*) const [virtual]

find the first axis with exactly the types

Parameters:

type

Returns:

an axis or null

14.27.4.15 virtual ConstAxisPtr MetNoFimex::CoordinateSystem::findAxisOfType (const std::vector< CoordinateAxis::AxisType > & *types*) const [virtual]

find the first axis with one of the types

Parameters:

types list of types

Returns:

an axis or null

14.27.4.16 virtual ConstAxisPtr MetNoFimex::CoordinateSystem::getGeoXAxis () const [virtual]

get the geographical x/lon-axis, that is one of GeoX, Longitude (in that order if several exist)

Returns:

an axis or null

14.27.4.17 virtual ConstAxisPtr MetNoFimex::CoordinateSystem::getGeoYAxis () const [virtual]

Set/overwrite the geographic y axis, that is one of GeoY, Latitude (in that order if several exist)

Returns:

an axis or null

14.27.4.18 `virtual ConstAxisPtr MetNoFimex::CoordinateSystem::getGeoZAxis () const`
[virtual]

get the geographical z-axis, that is one of GeoZ, Height, Pressure (in that order if several exist)

Returns:

an axis, or null

14.27.4.19 `virtual ConstAxisPtr MetNoFimex::CoordinateSystem::getTimeAxis () const`
[virtual]

get the time-axis, or NULL/0

14.27.4.20 `virtual ConstAxisList MetNoFimex::CoordinateSystem::getAxes () const` [virtual]

get all axes

14.27.4.21 `virtual void MetNoFimex::CoordinateSystem::setAxis (AxisPtr axis)` [virtual]

Set any axis.

Exceptions:

[CDMException](#) if an axis with the same axistype (except undefined) exists

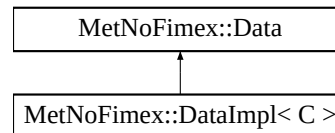
The documentation for this class was generated from the following file:

- include/fimex/coordSys/[CoordinateSystem.h](#)

14.28 MetNoFimex::Data Class Reference

```
#include <fimex/Data.h>
```

Inheritance diagram for MetNoFimex::Data::



Public Member Functions

- virtual `~Data()` =0
- virtual `size_t size()` const =0
size of the data
- virtual `int bytes_for_one()` const =0
sizeof the data-impl datatype
- virtual `void * getDataPtr()` =0
- virtual `void toStream(std::ostream &, std::string separator="")` const =0
printing of the current data to ostream, with optional separator
- virtual `const boost::shared_array< char > asConstChar()` const =0
retrieve data as char
- virtual `boost::shared_array< char > asChar()` =0
retrieve data as char
- virtual `const boost::shared_array< short > asConstShort()` const =0
retrieve data as short
- virtual `boost::shared_array< short > asShort()` =0
retrieve data as short
- virtual `const boost::shared_array< int > asConstInt()` const =0
retrieve data as int
- virtual `boost::shared_array< int > asInt()` =0
retrieve data as int
- virtual `const boost::shared_array< float > asConstFloat()` const =0
retrieve data as float
- virtual `boost::shared_array< float > asFloat()` =0
retrieve data as float (eventually copy)
- virtual `const boost::shared_array< double > asConstDouble()` const =0

retrieve data as double

- virtual `boost::shared_array< double > asDouble ()=0`
retrieve data as double
- virtual `std::string asString (std::string separator="") const =0`
retrieve the whole array as a string (with possible separator)
- virtual void `setValue (long pos, double val)=0`
set a value at the desired position
- virtual void `setValues (size_t startPos, const Data &data, size_t first=0, size_t end=-1)=0` throw (CDMEException)
- virtual void `setAllValues (double val)=0`
- virtual `boost::shared_ptr< Data > slice (std::vector< size_t > orgDimSize, std::vector< size_t > startDims, std::vector< size_t > outputDimSize)=0` throw (CDMEException)
get a multi-dimensional slice of the data
- virtual `boost::shared_ptr< Data > convertDataType (double oldFill, double oldScale, double oldOffset, CDMDataType newType, double newFill, double newScale, double newOffset)=0` throw (CDMEException)
convert the datatype from one type,fill,scale,offset to another
- virtual `CDMDataType getDataType () const =0`

14.28.1 Detailed Description

General class for storing different basic array pointers plus length

14.28.2 Constructor & Destructor Documentation

14.28.2.1 virtual `MetNoFimex::Data::~~Data ()` [pure virtual]

14.28.3 Member Function Documentation

14.28.3.1 virtual `size_t MetNoFimex::Data::size () const` [pure virtual]

size of the data

Implemented in `MetNoFimex::DataImpl< C >`.

14.28.3.2 virtual `int MetNoFimex::Data::bytes_for_one () const` [pure virtual]

sizeof the data-impl datatype

Implemented in `MetNoFimex::DataImpl< C >`.

14.28.3.3 virtual `void* MetNoFimex::Data::getDataPtr ()` [pure virtual]

Implemented in `MetNoFimex::DataImpl< C >`.

14.28.3.4 `virtual void MetNoFimex::Data::toStream (std::ostream &, std::string separator = "") const` [pure virtual]

printing of the current data to ostream, with optional separator

Implemented in [MetNoFimex::DataImpl< C >](#).

14.28.3.5 `virtual const boost::shared_array<char> MetNoFimex::Data::asConstChar () const` [pure virtual]

retrieve data as char

Implemented in [MetNoFimex::DataImpl< C >](#).

14.28.3.6 `virtual boost::shared_array<char> MetNoFimex::Data::asChar ()` [pure virtual]

retrieve data as char

Implemented in [MetNoFimex::DataImpl< C >](#).

14.28.3.7 `virtual const boost::shared_array<short> MetNoFimex::Data::asConstShort () const` [pure virtual]

retrieve data as short

Implemented in [MetNoFimex::DataImpl< C >](#).

14.28.3.8 `virtual boost::shared_array<short> MetNoFimex::Data::asShort ()` [pure virtual]

retrieve data as short

Implemented in [MetNoFimex::DataImpl< C >](#).

14.28.3.9 `virtual const boost::shared_array<int> MetNoFimex::Data::asConstInt () const` [pure virtual]

retrieve data as int

Implemented in [MetNoFimex::DataImpl< C >](#).

14.28.3.10 `virtual boost::shared_array<int> MetNoFimex::Data::asInt ()` [pure virtual]

retrieve data as int

Implemented in [MetNoFimex::DataImpl< C >](#).

14.28.3.11 `virtual const boost::shared_array<float> MetNoFimex::Data::asConstFloat () const` [pure virtual]

retrieve data as float

Implemented in [MetNoFimex::DataImpl< C >](#).

14.28.3.12 `virtual boost::shared_array<float> MetNoFimex::Data::asFloat () [pure virtual]`

retrieve data as float (eventually copy)

Implemented in [MetNoFimex::DataImpl< C >](#).

14.28.3.13 `virtual const boost::shared_array<double> MetNoFimex::Data::asConstDouble () const [pure virtual]`

retrieve data as double

Implemented in [MetNoFimex::DataImpl< C >](#).

14.28.3.14 `virtual boost::shared_array<double> MetNoFimex::Data::asDouble () [pure virtual]`

retrieve data as double

Implemented in [MetNoFimex::DataImpl< C >](#).

14.28.3.15 `virtual std::string MetNoFimex::Data::asString (std::string separator = " ") const [pure virtual]`

retrieve the whole array as a string (with possible separator)

Implemented in [MetNoFimex::DataImpl< C >](#).

14.28.3.16 `virtual void MetNoFimex::Data::setValue (long pos, double val) [pure virtual]`

set a value at the desired position

Implemented in [MetNoFimex::DataImpl< C >](#).

14.28.3.17 `virtual void MetNoFimex::Data::setValues (size_t startPos, const Data & data, size_t first = 0, size_t end = -1) throw (CDMException) [pure virtual]`

set the values from another [Data](#) implementation

Parameters:

startPos the first position the data should be written to

data the other data-source

first the first data-entry

end the last (excluded) data-entry, defaults to MAX size_t, automatically shrunk to fit size

Implemented in [MetNoFimex::DataImpl< C >](#), [MetNoFimex::DataImpl< C >](#), [MetNoFimex::DataImpl< C >](#), [MetNoFimex::DataImpl< C >](#), [MetNoFimex::DataImpl< C >](#), and [MetNoFimex::DataImpl< C >](#).

14.28.3.18 virtual void MetNoFimex::Data::setAllValues (double *val*) [pure virtual]

set all values to the submitted value

Parameters:

val value to set

Implemented in [MetNoFimex::DataImpl< C >](#).

14.28.3.19 virtual boost::shared_ptr<Data> MetNoFimex::Data::slice (std::vector< size_t > *orgDimSize*, std::vector< size_t > *startDims*, std::vector< size_t > *outputDimSize*) throw (CDMException) [pure virtual]

get a multi-dimensional slice of the data

This slices a multidimensional chunk out of the data. All parameters must be vectors of the same size (dimension of array). The first dimension is the fastest moving index (fortran arrays)

Parameters:

orgDimSize the dimensions of this vector. The product of all *orgDimSizes* must equal to *data.size*.

startDims The start-position in the original data to fetch data from

outputDimSize the size of the output data

Returns:

a [Data](#) of the size of *outputDimSize* with the same datatype as the original type

Exceptions:

[CDMException](#) on dimension mismatch: (*start+size* > *orgDimSize*) or (*Product(orgDimSize)* != *size*)

Implemented in [MetNoFimex::DataImpl< C >](#).

14.28.3.20 virtual boost::shared_ptr<Data> MetNoFimex::Data::convertDataType (double *oldFill*, double *oldScale*, double *oldOffset*, CDMDataType *newType*, double *newFill*, double *newScale*, double *newOffset*) throw (CDMException) [pure virtual]

convert the datatype from one type,fill,scale,offset to another

Implemented in [MetNoFimex::DataImpl< C >](#).

14.28.3.21 virtual CDMDataType MetNoFimex::Data::getDataType () const [pure virtual]

return the CDMDataType of this data

Implemented in [MetNoFimex::DataImpl< C >](#).

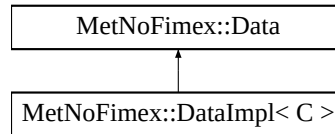
The documentation for this class was generated from the following file:

- [include/fimex/Data.h](#)

14.29 MetNoFimex::DataImpl< C > Class Template Reference

```
#include <DataImpl.h>
```

Inheritance diagram for MetNoFimex::DataImpl< C >::



Public Member Functions

- [DataImpl](#) (long length)
constructor where the array will be automatically allocated
- [DataImpl](#) (boost::shared_array< C > array, long length)
- virtual [~DataImpl](#) ()
- virtual size_t [size](#) () const
size of the data
- virtual int [bytes_for_one](#) () const
size of the data-impl datatype
- virtual void * [getDataPtr](#) ()
- virtual void [toStream](#) (std::ostream &os, std::string separator="") const
printing of the current data to ostream, with optional separator
- virtual const boost::shared_array< C > [asBase](#) () const
get the datapointer of the data
- template<typename T>
const boost::shared_array< T > [as](#) () const
- template<typename T>
boost::shared_array< T > [as](#) ()
- virtual const boost::shared_array< char > [asConstChar](#) () const
retrieve data as char
- virtual boost::shared_array< char > [asChar](#) ()
retrieve data as char
- virtual const boost::shared_array< short > [asConstShort](#) () const
retrieve data as short
- virtual boost::shared_array< short > [asShort](#) ()
retrieve data as short
- virtual const boost::shared_array< int > [asConstInt](#) () const
retrieve data as int

- virtual boost::shared_array< int > [asInt](#) ()
retrieve data as int
- virtual const boost::shared_array< float > [asConstFloat](#) () const
retrieve data as float
- virtual boost::shared_array< float > [asFloat](#) ()
retrieve data as float (eventually copy)
- virtual const boost::shared_array< double > [asConstDouble](#) () const
retrieve data as double
- virtual boost::shared_array< double > [asDouble](#) ()
retrieve data as double
- virtual **std::string** [asString](#) (**std::string** separator="") const
retrieve the whole array as a string (with possible separator)
- virtual void [setValue](#) (long pos, double val)
set a value at the desired position
- virtual void [setValues](#) (size_t startPos, const [Data](#) &data, size_t first=0, size_t last=-1) throw (CDMEException)
- virtual void [setAllValues](#) (double val)
- virtual boost::shared_ptr< [Data](#) > [slice](#) (**std::vector**< size_t > orgDimSize, **std::vector**< size_t > startDims, **std::vector**< size_t > outputDimSize) throw (CDMEException)
get a multi-dimensional slice of the data
- virtual boost::shared_ptr< [Data](#) > [convertDataType](#) (double oldFill, double oldScale, double oldOffset, [CDMDataType](#) newType, double newFill, double newScale, double newOffset) throw (CDMEException)
convert the datatype from one type,fill,scale,offset to another
- virtual [CDMDataType](#) [getDataType](#) () const
- template<class InputIterator>
void [setValues](#) (InputIterator begin, InputIterator end, size_t dataStartPos=0) throw (CDMEException)
- template<>
void [setValues](#) (size_t startPos, const [Data](#) &data, size_t first, size_t last) throw(CDMEException)
- template<>
void [setValues](#) (size_t startPos, const [Data](#) &data, size_t first, size_t last) throw(CDMEException)
- template<>
void [setValues](#) (size_t startPos, const [Data](#) &data, size_t first, size_t last) throw(CDMEException)
- template<>
void [setValues](#) (size_t startPos, const [Data](#) &data, size_t first, size_t last) throw(CDMEException)
- template<>
void [setValues](#) (size_t startPos, const [Data](#) &data, size_t first, size_t last) throw(CDMEException)

template<typename C> class MetNoFimex::DataImpl< C >

14.29.1 Constructor & Destructor Documentation

14.29.1.1 **template<typename C> MetNoFimex::DataImpl< C >::DataImpl (long *length*)**
[inline, explicit]

constructor where the array will be automatically allocated

14.29.1.2 **template<typename C> MetNoFimex::DataImpl< C >::DataImpl**
(boost::shared_array< C > *array*, long *length*) [inline, explicit]

14.29.1.3 **template<typename C> virtual MetNoFimex::DataImpl< C >::~~DataImpl ()**
[inline, virtual]

14.29.2 Member Function Documentation

14.29.2.1 **template<typename C> virtual size_t MetNoFimex::DataImpl< C >::size () const**
[inline, virtual]

size of the data

Implements [MetNoFimex::Data](#).

14.29.2.2 **template<typename C> virtual int MetNoFimex::DataImpl< C >::bytes_for_one ()**
const [inline, virtual]

sizeof the data-impl datatype

Implements [MetNoFimex::Data](#).

14.29.2.3 **template<typename C> virtual void* MetNoFimex::DataImpl< C >::getDataPtr ()**
[inline, virtual]

Implements [MetNoFimex::Data](#).

14.29.2.4 **template<typename C> void MetNoFimex::DataImpl< C >::toStream (std::ostream &,**
std::string *separator* = "") const [inline, virtual]

printing of the current data to ostream, with optional separator

Implements [MetNoFimex::Data](#).

Referenced by MetNoFimex::DataImpl< C >::asString().

14.29.2.5 **template<typename C> virtual const boost::shared_array<C>**
MetNoFimex::DataImpl< C >::asBase () const [inline, virtual]

get the datapointer of the data

14.29.2.6 `template<typename C> template<typename T> const boost::shared_array<T>
MetNoFimex::DataImpl< C >::as () const [inline]`

general conversion function, not in base since template methods not allowed

14.29.2.7 `template<typename C> template<typename T> boost::shared_array<T>
MetNoFimex::DataImpl< C >::as () [inline]`

14.29.2.8 `template<typename C> virtual const boost::shared_array<char>
MetNoFimex::DataImpl< C >::asConstChar () const [inline, virtual]`

retrieve data as char

Implements [MetNoFimex::Data](#).

14.29.2.9 `template<typename C> virtual boost::shared_array<char> MetNoFimex::DataImpl<
C >::asChar () [inline, virtual]`

retrieve data as char

Implements [MetNoFimex::Data](#).

14.29.2.10 `template<typename C> virtual const boost::shared_array<short>
MetNoFimex::DataImpl< C >::asConstShort () const [inline, virtual]`

retrieve data as short

Implements [MetNoFimex::Data](#).

14.29.2.11 `template<typename C> virtual boost::shared_array<short>
MetNoFimex::DataImpl< C >::asShort () [inline, virtual]`

retrieve data as short

Implements [MetNoFimex::Data](#).

14.29.2.12 `template<typename C> virtual const boost::shared_array<int>
MetNoFimex::DataImpl< C >::asConstInt () const [inline, virtual]`

retrieve data as int

Implements [MetNoFimex::Data](#).

14.29.2.13 `template<typename C> virtual boost::shared_array<int> MetNoFimex::DataImpl<
C >::asInt () [inline, virtual]`

retrieve data as int

Implements [MetNoFimex::Data](#).

14.29.2.14 `template<typename C> virtual const boost::shared_array<float>
MetNoFimex::DataImpl< C >::asConstFloat () const [inline, virtual]`

retrieve data as float

Implements [MetNoFimex::Data](#).

14.29.2.15 `template<typename C> virtual boost::shared_array<float> MetNoFimex::DataImpl<
C >::asFloat () [inline, virtual]`

retrieve data as float (eventually copy)

Implements [MetNoFimex::Data](#).

14.29.2.16 `template<typename C> virtual const boost::shared_array<double>
MetNoFimex::DataImpl< C >::asConstDouble () const [inline, virtual]`

retrieve data as double

Implements [MetNoFimex::Data](#).

14.29.2.17 `template<typename C> virtual boost::shared_array<double>
MetNoFimex::DataImpl< C >::asDouble () [inline, virtual]`

retrieve data as double

Implements [MetNoFimex::Data](#).

14.29.2.18 `template<typename C> std::string MetNoFimex::DataImpl< C >::asString
(std::string separator = " ") const [inline, virtual]`

retrieve the whole array as a string (with possible separator)

Implements [MetNoFimex::Data](#).

References `std::basic_ostringstream< _CharT, _Traits, _Alloc >::str()`, and `MetNoFimex::DataImpl< C >::toStream()`.

14.29.2.19 `template<typename C> virtual void MetNoFimex::DataImpl< C >::setValue (long
pos, double val) [inline, virtual]`

set a value at the desired position

Implements [MetNoFimex::Data](#).

14.29.2.20 `template<typename C> virtual void MetNoFimex::DataImpl< C >::setValues (size_t
startPos, const Data & data, size_t first = 0, size_t end = -1) throw (CDMException)
[virtual]`

set the values from another [Data](#) implementation

Parameters:

startPos the first position the data should be written to

data the other data-source

first the first data-entry

end the last (excluded) data-entry, defaults to MAX size_t, automatically shrunken to fit size

Implements [MetNoFimex::Data](#).

14.29.2.21 `template<typename C> virtual void MetNoFimex::DataImpl< C >::setAllValues
(double val) [inline, virtual]`

set all values to the submitted value

Parameters:

val value to set

Implements [MetNoFimex::Data](#).

14.29.2.22 `template<typename C> boost::shared_ptr< Data > MetNoFimex::DataImpl<
C >::slice (std::vector< size_t > orgDimSize, std::vector< size_t > startDims,
std::vector< size_t > outputDimSize) throw (CDMException) [inline, virtual]`

get a multi-dimensional slice of the data

This slices a multidimensional chunk out of the data. All parameters must be vectors of the same size (dimension of array). The first dimension is the fastest moving index (fortran arrays)

Parameters:

orgDimSize the dimensions of this vector. The product of all orgDimSizes must equal to data.size.

startDims The start-position in the original data to fetch data from

outputDimSize the size of the output data

Returns:

a [Data](#) of the size of outputDimSize with the same datatype as the original type

Exceptions:

[CDMException](#) on dimension mismatch: (start+size > orgDimSize) or (Product(orgDimSize) != size)

Implements [MetNoFimex::Data](#).

References [MetNoFimex::recursiveCopyMultiDimData\(\)](#), `std::vector< _Tp, _Alloc >::size()`, and [MetNoFimex::type2string\(\)](#).

14.29.2.23 `template<typename C> boost::shared_ptr< Data > MetNoFimex::DataImpl<
C >::convertDataType (double oldFill, double oldScale, double oldOffset,
CDMDataType newType, double newFill, double newScale, double newOffset) throw
(CDMException) [inline, virtual]`

convert the datatype from one type,fill,scale,offset to another

Implements [MetNoFimex::Data](#).

References [MetNoFimex::CDM_CHAR](#), [MetNoFimex::CDM_DOUBLE](#), [MetNoFimex::CDM_FLOAT](#), [MetNoFimex::CDM_INT](#), [MetNoFimex::CDM_NAT](#), [MetNoFimex::CDM_SHORT](#), and [MetNoFimex::CDM_STRING](#).

14.29.2.24 `template<typename C> virtual CDMDataType MetNoFimex::DataImpl< C >::getDataType () const [inline, virtual]`

return the CDMDataType of this data

Implements [MetNoFimex::Data](#).

References [MetNoFimex::CDM_NAT](#).

14.29.2.25 `template<typename C> template<class InputIterator> void MetNoFimex::DataImpl< C >::setValues (InputIterator begin, InputIterator end, size_t dataStartPos = 0) throw (CDMException) [inline]`

set the values of the data by the input-iterator

References [std::distance\(\)](#), [std::transform\(\)](#), and [MetNoFimex::type2string\(\)](#).

14.29.2.26 `template<> void MetNoFimex::DataImpl< char >::setValues (size_t startPos, const Data & data, size_t first, size_t end) throw(CDMException) [inline, virtual]`

set the values from another [Data](#) implementation

Parameters:

startPos the first position the data should be written to

data the other data-source

first the first data-entry

end the last (excluded) data-entry, defaults to MAX size_t, automatically shrunk to fit size

Implements [MetNoFimex::Data](#).

14.29.2.27 `template<> void MetNoFimex::DataImpl< short >::setValues (size_t startPos, const Data & data, size_t first, size_t end) throw(CDMException) [inline, virtual]`

set the values from another [Data](#) implementation

Parameters:

startPos the first position the data should be written to

data the other data-source

first the first data-entry

end the last (excluded) data-entry, defaults to MAX size_t, automatically shrunk to fit size

Implements [MetNoFimex::Data](#).

14.29.2.28 `template<> void MetNoFimex::DataImpl< int >::setValues (size_t startPos, const Data & data, size_t first, size_t end) throw(CDMException) [inline, virtual]`

set the values from another [Data](#) implementation

Parameters:

startPos the first position the data should be written to

data the other data-source

first the first data-entry

end the last (excluded) data-entry, defaults to MAX size_t, automatically shrunken to fit size

Implements [MetNoFimex::Data](#).

14.29.2.29 `template<> void MetNoFimex::DataImpl< float >::setValues (size_t startPos, const Data & data, size_t first, size_t end) throw(CDMException) [inline, virtual]`

set the values from another [Data](#) implementation

Parameters:

startPos the first position the data should be written to

data the other data-source

first the first data-entry

end the last (excluded) data-entry, defaults to MAX size_t, automatically shrunken to fit size

Implements [MetNoFimex::Data](#).

14.29.2.30 `template<> void MetNoFimex::DataImpl< double >::setValues (size_t startPos, const Data & data, size_t first, size_t end) throw(CDMException) [inline, virtual]`

set the values from another [Data](#) implementation

Parameters:

startPos the first position the data should be written to

data the other data-source

first the first data-entry

end the last (excluded) data-entry, defaults to MAX size_t, automatically shrunken to fit size

Implements [MetNoFimex::Data](#).

The documentation for this class was generated from the following file:

- include/fimex/[DataImpl.h](#)

14.30 MetNoFimex::DataTypeChanger Class Reference

```
#include <DataTypeChanger.h>
```

Public Member Functions

- [DataTypeChanger](#) ([CDMDataType](#) oldType)
- [DataTypeChanger](#) ([CDMDataType](#) oldType, double oldFill, double oldScale, double oldOffset, [CDMDataType](#) newType, double newFill, double newScale, double newOffset, double unitScale=1., double unitOffset=0.)
- virtual [~DataTypeChanger](#) ()
- boost::shared_ptr< [Data](#) > [convertData](#) (boost::shared_ptr< [Data](#) >) const throw (CDMException)
- [CDMDataType](#) [getDataType](#) () const

14.30.1 Detailed Description

brief wrapper class around data->convertType

14.30.2 Constructor & Destructor Documentation

14.30.2.1 MetNoFimex::DataTypeChanger::DataTypeChanger ([CDMDataType](#) *oldType*) [explicit]

initialize data with the oldType convertData will do nothing in this case

14.30.2.2 MetNoFimex::DataTypeChanger::DataTypeChanger ([CDMDataType](#) *oldType*, double *oldFill*, double *oldScale*, double *oldOffset*, [CDMDataType](#) *newType*, double *newFill*, double *newScale*, double *newOffset*, double *unitScale* = 1., double *unitOffset* = 0.) [explicit]

initialize with the old and new settings

Parameters:

oldType datatype of original data

oldFill fill value of the original data

oldScale scale_factor of the original data

oldOffset scale_factor of the original data

newType datatype of converted data

newFill fill value of converted data

newScale scale_factor of the converted data

newOffset add_offset of the converted data

unitScale scale_factor for the unpacked data, i.e. for unit changes, default 1.

unitOffset offset for the unpacked data, i.e. for unit changes, default 0.

14.30.2.3 virtual MetNoFimex::DataTypeChanger::~~DataTypeChanger () [virtual]

14.30.3 Member Function Documentation

14.30.3.1 boost::shared_ptr<Data> MetNoFimex::DataTypeChanger::convertData
(boost::shared_ptr< Data >) const throw (CDMException)

convert the data to the new scale/fill/offset

14.30.3.2 CDMDataType MetNoFimex::DataTypeChanger::getDataType () const

return the datatype of the converted data

The documentation for this class was generated from the following file:

- include/fimex/[DataTypeChanger.h](#)

14.31 MetNoFelt::Felt_Array Class Reference

encapsulate parameters of a [felt](#) file

```
#include <Felt_Array.h>
```

Public Member Functions

- [Felt_Array](#) ()
- [Felt_Array](#) (const **string** name, const boost::array< short, 16 > idx, const **string** &dataType="short")
- virtual [~Felt_Array](#) ()
- void [addInformationByIndex](#) (const boost::array< short, 16 > &idx, int fieldSize) throw (Felt_File_Error)
- const boost::array< short, 16 > & [getIndexHeader](#) () const
get the time/level independent index of this header
- void [setDataHeader](#) (boost::array< short, 20 > header) throw (Felt_File_Error)
set the [felt](#) data-header for this array a [Felt_File_Error](#) will be thrown if the header is different for the different times/layers of this Array
- const boost::array< short, 20 > & [getDataHeader](#) () const
get the time/level independent data-header
- short [getLevelType](#) () const
get the [felt](#) level type of this array
- void [setGridType](#) (int gridType)
set the gridType as used in libmi gridPar function
- int [getGridType](#) () const
get the gridType
- void [setGridParameters](#) (boost::array< float, 6 > gridParameters)
- const boost::array< float, 6 > & [getGridParameters](#) () const
get the extra grid information from the end of the data
- const **string** & [getName](#) () const
- const **string** & [getDatatype](#) () const
- double [getFillValue](#) () const
- void [setFillValue](#) (double fillValue)
- **vector**< epoch_seconds > [getTimes](#) () const
- **vector**< short > [getLevels](#) () const
- **vector**< **pair**< short, short > > [getLevelPairs](#) () const
- void [addIdent19](#) (epoch_seconds time, **pair**< short, short > levelPair, short value)
- short [getIdent19](#) (epoch_seconds time, **pair**< short, short > levelPair) const throw (Felt_File_Error)
- short [getIdent19](#) (**pair**< short, short > levelPair) const throw (Felt_File_Error)
- short [getIdent19](#) (epoch_seconds time) const throw (Felt_File_Error)
- short [getIdent19](#) () const throw (Felt_File_Error)
- int [getX](#) () const

- int [getY](#) () const
- short [getVerticalFeltType](#) () const
- double [getScalingFactor](#) () const
- boost::array< short, 16 > const [getIndex](#) (epoch_seconds time, short level) throw (Felt_File_Error)
- int [getFieldSize](#) (epoch_seconds time, short level) const throw (Felt_File_Error)

14.31.1 Detailed Description

encapsulate parameters of a [felt](#) file

store local variables of a parameter, partially retrieved from the diana.setup, partially retrieved from the file

14.31.2 Constructor & Destructor Documentation

14.31.2.1 MetNoFelt::Felt_Array::Felt_Array ()

constructor

14.31.2.2 MetNoFelt::Felt_Array::Felt_Array (const string *name*, const boost::array< short, 16 > *idx*, const string & *dataType* = "short") [explicit]

constructor applying the parameter name and the [felt](#) description index array

Parameters:

name parameter name

idx feltfiles are indexed by a short[16] arrays representing different parameters. The parameters used here are those applied to qfelt (query [felt](#))

dataType short|float|double datatype used for autoscaling, [getScalingFactor\(\)](#) will be allways return 1 for float and double

14.31.2.3 virtual MetNoFelt::Felt_Array::~~Felt_Array () [virtual]

14.31.3 Member Function Documentation

14.31.3.1 void MetNoFelt::Felt_Array::addInformationByIndex (const boost::array< short, 16 > & *idx*, int *fieldSize*) throw (Felt_File_Error)

add information from the felt-index (usually retrieved from qfelt) to this [Felt_Array](#) the index given here must correspond to the initialization index

14.31.3.2 const boost::array<short, 16> & MetNoFelt::Felt_Array::getIndexHeader () const [inline]

get the time/level independent index of this header

14.31.3.3 void MetNoFelt::Felt_Array::setDataHeader (boost::array< short, 20 > *header*) throw (Felt_File_Error)

set the [felt](#) data-header for this array a [Felt_File_Error](#) will be thrown if the header is different for the different times/layers of this Array

Exceptions:

[Felt_File_Error](#) when data-definitions change

14.31.3.4 const boost::array<short, 20>& MetNoFelt::Felt_Array::getDataHeader () const [inline]

get the time/level independent data-header

14.31.3.5 short MetNoFelt::Felt_Array::getLevelType () const [inline]

get the [felt](#) level type of this array

14.31.3.6 void MetNoFelt::Felt_Array::setGridType (int *gridType*) [inline]

set the gridType as used in libmi gridPar function

14.31.3.7 int MetNoFelt::Felt_Array::getGridType () const [inline]

get the gridType

14.31.3.8 void MetNoFelt::Felt_Array::setGridParameters (boost::array< float, 6 > *gridParameters*) [inline]

set all the grid parameters from the [felt](#) file as retrieved from libmi's gridPar function

14.31.3.9 const boost::array<float, 6>& MetNoFelt::Felt_Array::getGridParameters () const [inline]

get the extra grid information from the end of the data

14.31.3.10 const string& MetNoFelt::Felt_Array::getName () const

return the parameter name

14.31.3.11 const string& MetNoFelt::Felt_Array::getDatatype () const [inline]

return the datatype as string short|float|double

14.31.3.12 double MetNoFelt::Felt_Array::getFillValue () const [inline]

return the changed fill used in Felt_File::getScaledDataSlice

14.31.3.13 void MetNoFelt::Felt_Array::setFillValue (double *fillValue*) [inline]

set the fill value to be used in Felt_File::getScaledDataSlice

14.31.3.14 vector<epoch_seconds> MetNoFelt::Felt_Array::getTimes () const

return the times available for this parameter, sorted

14.31.3.15 vector<short> MetNoFelt::Felt_Array::getLevels () const

return the levels available for this parameter, sorted

14.31.3.16 vector<pair<short, short> > MetNoFelt::Felt_Array::getLevelPairs () const

return the level pairs (niveau 1, niveau 2) for this parameter as used by hybrid levels

14.31.3.17 void MetNoFelt::Felt_Array::addIdent19 (epoch_seconds *time*, pair< short, short > *levelPair*, short *value*) [inline]

add the ident19 parameter from the data-header

14.31.3.18 short MetNoFelt::Felt_Array::getIdent19 (epoch_seconds *time*, pair< short, short > *levelPair*) const throw (Felt_File_Error)

get the ident19 parameter from the data-header, throw error if levelPair/time doesn't exists

Warning:

only ident19 of data already read will be taken into account

14.31.3.19 short MetNoFelt::Felt_Array::getIdent19 (pair< short, short > *levelPair*) const throw (Felt_File_Error)

get the ident19 parameter from the data-header, assures that the parameters keep constant across all times for each levelPair or throws a [Felt_File_Error](#)

Warning:

only ident19 of data already read will be taken into account

14.31.3.20 short MetNoFelt::Felt_Array::getIdent19 (epoch_seconds *time*) const throw (Felt_File_Error)

get the ident19 parameter from the data-header, assures that the parameters keep constant across all level-Pair for each time or throws a [Felt_File_Error](#)

Warning:

only ident19 of data already read will be taken into account

14.31.3.21 short MetNoFelt::Felt_Array::getIdent19 () const throw (Felt_File_Error)

get the ident19 parameter from the data-header, assures that the parameters keep constant across all level-Pair and times or throws a [Felt_File_Error](#)

Warning:

only ident19 of data already read will be taken into account

14.31.3.22 int MetNoFelt::Felt_Array::getX () const [inline]

return x/longitude size

14.31.3.23 int MetNoFelt::Felt_Array::getY () const [inline]

return y/latitude size

14.31.3.24 short MetNoFelt::Felt_Array::getVerticalFeltType () const [inline]

return the felt-type of the vertical axis

14.31.3.25 double MetNoFelt::Felt_Array::getScalingFactor () const

return scalingFactor

14.31.3.26 boost::array<short, 16> const MetNoFelt::Felt_Array::getIndex (epoch_seconds *time*, short *level*) throw (Felt_File_Error)

return a copy of the index used within this [Felt_Array](#)

14.31.3.27 int MetNoFelt::Felt_Array::getFieldSize (epoch_seconds *time*, short *level*) const throw (Felt_File_Error)

The documentation for this class was generated from the following file:

- include/fimex/[Felt_Array.h](#)

14.32 MetNoFelt::Felt_Array2 Class Reference

A collection of FeltFields to build a 4-dimensional data array for 1-parameter and 1-vertical coordinate.

```
#include <Felt_Array2.h>
```

Public Member Functions

- [Felt_Array2](#) (const **string** name, const boost::shared_ptr< [felt::FeltField](#) > feltField, const **string** &dataType, double fillValue)
- virtual [~Felt_Array2](#) ()
- void [addInformationByField](#) (boost::shared_ptr< [felt::FeltField](#) > field) throw (Felt_File_Error)
- const **string** & [getName](#) () const
get the time/level independent data-header
- const **string** & [getDatatype](#) () const
- int [getGrid](#) (boost::posix_time::ptime time, **LevelPair** levelPair, **vector**< short > &gridOut) throw (Felt_File_Error)
- int [getGridAllowDelta](#) (boost::posix_time::ptime time, **LevelPair** levelPair, **vector**< short > &gridOut, const boost::array< float, 6 > &gridParameterDelta) throw (Felt_File_Error)
- int [getLevelType](#) () const
get the [felt](#) level type of this array
- double [getFillValue](#) () const
- **vector**< boost::posix_time::ptime > [getTimes](#) () const
- **vector**< **LevelPair** > [getLevelPairs](#) () const
- int [getIdent19](#) (boost::posix_time::ptime time, **LevelPair** levelPair) const throw (Felt_File_Error)
- int [getX](#) () const
- int [getY](#) () const
- int [scaleFactor](#) () const
- double [getScalingFactor](#) () const
- boost::shared_ptr< [felt::FeltGridDefinition](#) > [getGridDefinition](#) () const
- int [getGridType](#) () const
- const boost::shared_ptr< [felt::FeltField](#) > [getField](#) (boost::posix_time::ptime time, **LevelPair** levelPair) const throw (Felt_File_Error)

14.32.1 Detailed Description

A collection of FeltFields to build a 4-dimensional data array for 1-parameter and 1-vertical coordinate.

A [Felt_Array2](#) collects all [felt::FeltField](#) from a [felt::FeltFile](#) with the same parameter and vertical coordinate. It is possible to access the data of such each layer.

14.32.2 Constructor & Destructor Documentation

14.32.2.1 MetNoFelt::Felt_Array2::Felt_Array2 (const string name, const boost::shared_ptr< [felt::FeltField](#) > *feltField*, const string & *dataType*, double *fillValue*) [explicit]

constructor applying the parameter name and the [felt](#) description from the first feltField

Parameters:

name parameter name

field a field added to this constructor

dataType short|float|double datatype used for autoscaling, [getScalingFactor\(\)](#) will be always return 1 for float and double

fillValue fillValue of the datatype, usually -32767

14.32.2.2 `virtual MetNoFelt::Felt_Array2::~~Felt_Array2 ()` [virtual]

14.32.3 Member Function Documentation

14.32.3.1 `void MetNoFelt::Felt_Array2::addInformationByField (boost::shared_ptr< felt::FeltField > field) throw (Felt_File_Error)`

add information from the felt-index (usually retrieved from qfelt) to this [Felt_Array2](#) the index given here must correspond to the initialization index

14.32.3.2 `const string& MetNoFelt::Felt_Array2::getName () const`

get the time/level independent data-header

return the parameter name

14.32.3.3 `const string& MetNoFelt::Felt_Array2::getDatatype () const` [inline]

return the datatype as string short|float|double

14.32.3.4 `int MetNoFelt::Felt_Array2::getGrid (boost::posix_time::ptime time, LevelPair levelPair, vector< short > & gridOut) throw (Felt_File_Error)`

read a grid for a time and a levelPair

Parameters:

time The time of the field

levelPair The levelPair of the field

gridOut The data of this field will be put into this grid

Returns:

the scaleFactor as tenth exponent of this field (grid * 10^scaleFactor)

Exceptions:

[Felt_File_Error](#) if the gridDefinition (gridType or gridParameters) change

14.32.3.5 `int MetNoFelt::Felt_Array2::getGridAllowDelta (boost::posix_time::ptime time, LevelPair levelPair, vector< short > & gridOut, const boost::array< float, 6 > & gridParameterDelta) throw (Felt_File_Error)`

same as getGrid, but the gridParameters to change up to the value provided in gridParameterDelta

14.32.3.6 int MetNoFelt::Felt_Array2::getLevelType () const

get the [felt](#) level type of this array

14.32.3.7 double MetNoFelt::Felt_Array2::getFillValue () const [inline]

return the changed fill used in Felt_File::getScaledDataSlice

14.32.3.8 vector<boost::posix_time::ptime> MetNoFelt::Felt_Array2::getTimes () const

return the times available for this parameter, sorted

14.32.3.9 vector<LevelPair> MetNoFelt::Felt_Array2::getLevelPairs () const

return the level pairs (niveau 1, niveau 2) for this parameter as used by hybrid levels

14.32.3.10 int MetNoFelt::Felt_Array2::getIdent19 (boost::posix_time::ptime *time*, LevelPair *levelPair*) const throw (Felt_File_Error)

get the ident19 parameter from the data-header, throw error if levelPair/time doesn't exists

Warning:

only ident19 of data already read will be taken into account

14.32.3.11 int MetNoFelt::Felt_Array2::getX () const

return x/longitude size

14.32.3.12 int MetNoFelt::Felt_Array2::getY () const

return y/latitude size

14.32.3.13 int MetNoFelt::Felt_Array2::scaleFactor () const

get the files scaleFactor, this corresponds to scalingFactor by $10^{(\text{scaleFactor})} == \text{scalingFactor}$

14.32.3.14 double MetNoFelt::Felt_Array2::getScalingFactor () const

return scalingFactor

14.32.3.15 `boost::shared_ptr<felt::FeltGridDefinition> MetNoFelt::Felt_Array2::getGridDefinition () const`

14.32.3.16 `int MetNoFelt::Felt_Array2::getGridType () const`

14.32.3.17 `const boost::shared_ptr<felt::FeltField> MetNoFelt::Felt_Array2::getField (boost::posix_time::ptime time, LevelPair levelPair) const throw (Felt_File_Error)`

The documentation for this class was generated from the following file:

- `include/fimex/Felt\_Array2.h`

14.33 MetNoFelt::Felt_File Class Reference

Felt File access.

```
#include <Felt_File.h>
```

Public Member Functions

- [Felt_File](#) ()
constructor
- [Felt_File](#) (const **std::string** &filename) throw (Felt_File_Error)
- [Felt_File](#) (const **std::string** &filename, const **std::vector**< **std::string** > &dianaParamList, const **std::map**< **std::string**, **std::string** > &options) throw (Felt_File_Error)
- virtual [~Felt_File](#) ()
- [Felt_Array](#) & [getFeltArray](#) (const **std::string** &compName) throw (Felt_File_Error)
retrieve a [Felt_Array](#)
- **std::vector**< short > [getDataSlice](#) (const **std::string** &compName, const epoch_seconds time, const short level) throw (Felt_File_Error)
retrieve a data slice
- boost::shared_ptr< [MetNoFimex::Data](#) > [getScaledDataSlice](#) (const **std::string** &compName, const epoch_seconds time, const short level, double fillValue) throw (Felt_File_Error)
- **std::vector**< [Felt_Array](#) > [listFeltArrays](#) ()
- **std::map**< short, **std::vector**< short > > [getFeltLevels](#) () const
- **std::map**< short, **std::vector**< **pair**< short, short > > > [getFeltLevelPairs](#) () const
- const **ShortPairMap** & [getHybridLevels](#) () const
- **std::vector**< epoch_seconds > [getFeltTimes](#) () const
all time values, sorted
- int [getNX](#) () const
get size in x direction
- int [getNY](#) () const
get size in y direction
- boost::shared_ptr< [MetNoFimex::Data](#) > [getXData](#) () const throw (Felt_File_Error)
get the values of the x axis
- boost::shared_ptr< [MetNoFimex::Data](#) > [getYData](#) () const throw (Felt_File_Error)
get the values of the y axis
- short [getGridType](#) () const throw (Felt_File_Error)
assumes one set of grid-type for the whole file
- const boost::array< float, 6 > & [getGridParameters](#) () const throw (Felt_File_Error)
assumes one set of grid-parameters for the whole file

14.33.1 Detailed Description

Felt File access.

[Felt_File](#) gives c++ style access to [felt](#) files. It uses internally libmi and caches the table of contents

14.33.2 Constructor & Destructor Documentation

14.33.2.1 MetNoFelt::Felt_File::Felt_File () [inline]

constructor

open an empty [felt](#) file, just a default constructor, no useful information

14.33.2.2 MetNoFelt::Felt_File::Felt_File (const std::string & *filename*) throw (Felt_File_Error) [explicit]

open and read toc of a [felt](#) file

Parameters:

filename name of [felt](#) file

14.33.2.3 MetNoFelt::Felt_File::Felt_File (const std::string & *filename*, const std::vector< std::string > & *dianaParamList*, const std::map< std::string, std::string > & *options*) throw (Felt_File_Error) [explicit]

open and read toc of a [felt](#) file

Parameters:

paramList a list of known parameters (in diana format, e.g. 17,2,1000:prod=74), only the known parameters will be read

Warning:

The diana format is extended by dataType=short|float|double and fillValue=(number in short|float|double) to add the return type of the data. Autoscaling will be turned on for 'get-DataSlice'. default is dataType=short:fillValue=-32767

14.33.2.4 virtual MetNoFelt::Felt_File::~~Felt_File () [virtual]

14.33.3 Member Function Documentation

14.33.3.1 Felt_Array& MetNoFelt::Felt_File::getFeltArray (const std::string & *compName*) throw (Felt_File_Error)

retrieve a [Felt_Array](#)

Parameters:

compName parameter name of [felt](#) file as named in diana setup

14.33.3.2 `std::vector<short> MetNoFelt::Felt_File::getDataSlice (const std::string & compName, const epoch_seconds time, const short level) throw (Felt_File_Error)`

retrieve a data slice

Parameters:

compName parameter name of [felt](#) file
time time of slice
level level of slice

14.33.3.3 `boost::shared_ptr<MetNoFimex::Data> MetNoFelt::Felt_File::getScaledDataSlice (const std::string & compName, const epoch_seconds time, const short level, double fillValue) throw (Felt_File_Error)`

retrieve the data prescaled (if float or double) and replaced with the new fill value

Parameters:

compName parameter name of [felt](#) file
time time of slice
level level of slice

14.33.3.4 `std::vector<Felt_Array> MetNoFelt::Felt_File::listFeltArrays ()`

retrieve all [felt](#) arrays

14.33.3.5 `std::map<short, std::vector<short> > MetNoFelt::Felt_File::getFeltLevels () const`

Z-axis types and values

Returns:

map consisting of [felt](#) level-ids and a sorted vector of level values

14.33.3.6 `std::map<short, std::vector<pair<short,short> > > MetNoFelt::Felt_File::getFeltLevelPairs () const`

Z-axis types and values

Returns:

map consisting of [felt](#) level-ids and a sorted vector of level-pairs of values

14.33.3.7 `const ShortPairMap& MetNoFelt::Felt_File::getHybridLevels () const` `[inline]`

14.33.3.8 `std::vector<epoch_seconds> MetNoFelt::Felt_File::getFeltTimes () const`

all time values, sorted

14.33.3.9 int MetNoFelt::Felt_File::getNX () const

get size in x direction

14.33.3.10 int MetNoFelt::Felt_File::getNY () const

get size in y direction

**14.33.3.11 boost::shared_ptr<MetNoFimex::Data> MetNoFelt::Felt_File::getXData () const
throw (Felt_File_Error)**

get the values of the x axis

**14.33.3.12 boost::shared_ptr<MetNoFimex::Data> MetNoFelt::Felt_File::getYData () const
throw (Felt_File_Error)**

get the values of the y axis

14.33.3.13 short MetNoFelt::Felt_File::getGridType () const throw (Felt_File_Error)

assumes one set of grid-type for the whole file

**14.33.3.14 const boost::array<float, 6>& MetNoFelt::Felt_File::getGridParameters () const
throw (Felt_File_Error)**

assumes one set of grid-parameters for the whole file

The documentation for this class was generated from the following file:

- include/fimex/[Felt_File.h](#)

14.34 MetNoFelt::Felt_File2 Class Reference

Felt File access.

```
#include <Felt_File2.h>
```

Public Member Functions

- [Felt_File2](#) ()
constructor
- [Felt_File2](#) (const **std::string** &filename) throw (Felt_File_Error)
- [Felt_File2](#) (const **std::string** &filename, const **std::vector**< **std::string** > &dianaParamList, const **std::map**< **std::string**, **std::string** > &options) throw (Felt_File_Error)
- virtual ~[Felt_File2](#) ()
- const boost::shared_ptr< [Felt_Array2](#) > [getFeltArray](#) (const **std::string** &compName) const throw (Felt_File_Error)
retrieve a [Felt_Array2](#)
- boost::shared_ptr< [MetNoFimex::Data](#) > [getScaledDataSlice](#) (boost::shared_ptr< [Felt_Array2](#) > feltArray, const boost::posix_time::ptime time, const **LevelPair** level) throw (Felt_File_Error)
retrieve a data slice
- **std::vector**< boost::shared_ptr< [Felt_Array2](#) > > [listFeltArrays](#) () const
- **std::map**< short, **std::vector**< **LevelPair** > > [getFeltLevelPairs](#) () const
- const **std::map**< **LevelPair**, int > & [getHybridLevels](#) () const
- **std::vector**< boost::posix_time::ptime > [getFeltTimes](#) () const
all time values, sorted
- int [getNX](#) () const
get size in x direction
- int [getNY](#) () const
get size in y direction
- boost::shared_ptr< [MetNoFimex::Data](#) > [getXData](#) () const throw (Felt_File_Error)
get the values of the x axis
- boost::shared_ptr< [MetNoFimex::Data](#) > [getYData](#) () const throw (Felt_File_Error)
get the values of the y axis
- int [getGridType](#) () const throw (Felt_File_Error)
- boost::shared_ptr< [felt::FeltGridDefinition](#) > [getGridDefinition](#) () const throw (Felt_File_Error)
assumes one set of grid-parameters for the whole file

14.34.1 Detailed Description

Felt File access.

[Felt_File2](#) gives c++ style access to [felt](#) files. It uses internally libmi and caches the table of contents

14.34.2 Constructor & Destructor Documentation

14.34.2.1 MetNoFelt::Felt_File2::Felt_File2 () [inline]

constructor

open an empty [felt](#) file, just a default constructor, no useful information

14.34.2.2 MetNoFelt::Felt_File2::Felt_File2 (const std::string & *filename*) throw (Felt_File_Error) [explicit]

open and read toc of a [felt](#) file

Parameters:

filename name of [felt](#) file

14.34.2.3 MetNoFelt::Felt_File2::Felt_File2 (const std::string & *filename*, const std::vector< std::string > & *dianaParamList*, const std::map< std::string, std::string > & *options*) throw (Felt_File_Error) [explicit]

open and read toc of a [felt](#) file

Parameters:

paramList a list of known parameters (in diana format, e.g. 17,2,1000:prod=74), only the known parameters will be read

Warning:

The diana format is extended by `dataType=short|float|double` and `fillValue=(number in short|float|double)` to add the return type of the data. Autoscaling will be turned on for 'get-DataSlice'. default is `dataType=short:fillValue=-32767`

14.34.2.4 virtual MetNoFelt::Felt_File2::~~Felt_File2 () [virtual]

14.34.3 Member Function Documentation

14.34.3.1 const boost::shared_ptr<Felt_Array2> MetNoFelt::Felt_File2::getFeltArray (const std::string & *compName*) const throw (Felt_File_Error)

retrieve a [Felt_Array2](#)

Parameters:

compName parameter name of [felt](#) file as named in diana setup

14.34.3.2 boost::shared_ptr<MetNoFimex::Data> MetNoFelt::Felt_File2::getScaledDataSlice (boost::shared_ptr< Felt_Array2 > *feltArray*, const boost::posix_time::ptime *time*, const LevelPair *level*) throw (Felt_File_Error)

retrieve a data slice

retrieve the data prescaled (if float or double) and replaced with the new fill value

Parameters:

compName parameter name of [felt](#) file

time time of slice

level level of slice

14.34.3.3 `std::vector<boost::shared_ptr<Felt_Array2> > MetNoFelt::Felt_File2::listFeltArrays () const`

retrieve all [felt](#) arrays

14.34.3.4 `std::map<short, std::vector<LevelPair> > MetNoFelt::Felt_File2::getFeltLevelPairs () const`

Z-axis types and values

Returns:

map consisting of [felt](#) level-ids and a sorted vector of level values Z-axis types and values

map consisting of [felt](#) level-ids and a sorted vector of level-pairs of values

14.34.3.5 `const std::map<LevelPair, int>& MetNoFelt::Felt_File2::getHybridLevels () const [inline]`

14.34.3.6 `std::vector<boost::posix_time::ptime> MetNoFelt::Felt_File2::getFeltTimes () const`

all time values, sorted

14.34.3.7 `int MetNoFelt::Felt_File2::getNX () const`

get size in x direction

14.34.3.8 `int MetNoFelt::Felt_File2::getNY () const`

get size in y direction

14.34.3.9 `boost::shared_ptr<MetNoFimex::Data> MetNoFelt::Felt_File2::getXData () const throw (Felt_File_Error)`

get the values of the x axis

14.34.3.10 `boost::shared_ptr<MetNoFimex::Data> MetNoFelt::Felt_File2::getYData () const throw (Felt_File_Error)`

get the values of the y axis

14.34.3.11 int MetNoFelt::Felt_File2::getGridType () const throw (Felt_File_Error)

assumes one set of grid-parameters for the whole file, returns parameter between 1 and 6, without extra definition

14.34.3.12 boost::shared_ptr<felt::FeltGridDefinition> MetNoFelt::Felt_File2::getGridDefinition () const throw (Felt_File_Error)

assumes one set of grid-parameters for the whole file

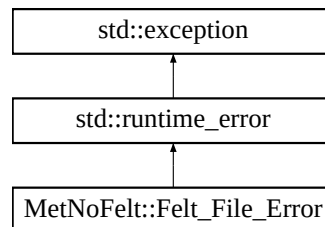
The documentation for this class was generated from the following file:

- include/fimex/[Felt_File2.h](#)

14.35 MetNoFelt::Felt_File_Error Class Reference

```
#include <Felt_File_Error.h>
```

Inheritance diagram for MetNoFelt::Felt_File_Error::



Public Member Functions

- [Felt_File_Error](#) (const `std::string` &message)

14.35.1 Constructor & Destructor Documentation

14.35.1.1 `MetNoFelt::Felt_File_Error::Felt_File_Error (const std::string & message)` [`inline`, `explicit`]

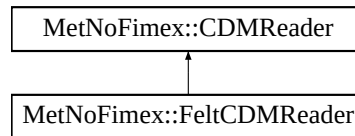
The documentation for this class was generated from the following file:

- `include/fimex/Felt\_File\_Error.h`

14.36 MetNoFimex::FeltCDMReader Class Reference

```
#include <FeltCDMReader.h>
```

Inheritance diagram for MetNoFimex::FeltCDMReader::



Public Member Functions

- [FeltCDMReader](#) (**std::string** filename, **std::string** configFilename) throw (CDMException)
- virtual [~FeltCDMReader](#) ()
- virtual boost::shared_ptr<[Data](#)> [getDataSlice](#) (const **std::string** &varName, size_t unLimDimPos) throw (CDMException)

14.36.1 Constructor & Destructor Documentation

14.36.1.1 MetNoFimex::FeltCDMReader::FeltCDMReader (**std::string** *filename*, **std::string** *configFilename*) throw (CDMException)

14.36.1.2 virtual MetNoFimex::FeltCDMReader::~~FeltCDMReader () [virtual]

14.36.2 Member Function Documentation

14.36.2.1 virtual boost::shared_ptr<[Data](#)> MetNoFimex::FeltCDMReader::getDataSlice (const **std::string** & *varName*, size_t *unLimDimPos*) throw (CDMException) [virtual]

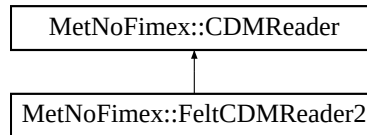
The documentation for this class was generated from the following file:

- include/fimex/[FeltCDMReader.h](#)

14.37 MetNoFimex::FeltCDMReader2 Class Reference

```
#include <FeltCDMReader2.h>
```

Inheritance diagram for MetNoFimex::FeltCDMReader2::



Public Member Functions

- [FeltCDMReader2](#) (**std::string** filename, **std::string** configFilename) throw (CDMException)
- virtual [~FeltCDMReader2](#) ()
- virtual boost::shared_ptr<[Data](#)> [getDataSlice](#) (const **std::string** &varName, size_t unLimDimPos) throw (CDMException)
data-reading function to be called from the [CDMWriter](#)

14.37.1 Constructor & Destructor Documentation

14.37.1.1 MetNoFimex::FeltCDMReader2::FeltCDMReader2 (**std::string** filename, **std::string** configFilename) throw (CDMException)

14.37.1.2 virtual MetNoFimex::FeltCDMReader2::~~FeltCDMReader2 () [virtual]

14.37.2 Member Function Documentation

14.37.2.1 virtual boost::shared_ptr<[Data](#)> MetNoFimex::FeltCDMReader2::getDataSlice (const **std::string** & varName, size_t unLimDimPos) throw (CDMException) [virtual]

data-reading function to be called from the [CDMWriter](#)

This methods needs to be implemented by the [CDMReader](#). It should provide the data for each variable, either by reading from disk, converting from another [CDMReader](#) or reading from an in-memory data-section.

This function should retrieve the whole data for a dataset without unlimited dimension if the unLimDimPos == 0.

Parameters:

varName name of the variable to read

unLimDimPos (optional) if the variable contains a unlimited dimension (max one allowed) an slice of this position is returned

Implements [MetNoFimex::CDMReader](#).

The documentation for this class was generated from the following file:

- include/fimex/[FeltCDMReader2.h](#)

14.38 felt::FeltField Class Reference

```
#include <FeltField.h>
```

Public Types

- typedef boost::array< [word](#), 16 > [Header](#)

Public Member Functions

- [FeltField](#) (const [FeltFile](#) &ff, size_t index)
- [~FeltField](#) ()
- bool [valid](#) () const
- int [producer](#) () const
- int [gridArea](#) () const
- boost::posix_time::ptime [referenceTime](#) () const
- boost::posix_time::ptime [validTime](#) () const
- int [parameter](#) () const
- int [dataType](#) () const
- int [verticalCoordinate](#) () const
- int [level1](#) () const
- int [level2](#) () const
- int [gridType](#) () const
- void [grid](#) (std::vector< [word](#) > &out) const
- size_t [gridSize](#) () const
- int [scaleFactor](#) () const
- int [xNum](#) () const
- int [yNum](#) () const
- int [miscField](#) () const

this field is described in the [felt](#) documentation as "word 19 in data part"

- int [dataVersion](#) () const
- [FeltGridDefinitionPtr](#) [projectionInformation](#) () const

throws

- std::string [information](#) () const
- std::string [gridInformation](#) () const
- const [Header](#) & [getHeader](#) () const

access [felt](#) index header

14.38.1 Member Typedef Documentation

14.38.1.1 `typedef boost::array<word, 16> felt::FeltField::Header`

14.38.2 Constructor & Destructor Documentation

14.38.2.1 `felt::FeltField::FeltField (const FeltFile &ff, size_t index)`

14.38.2.2 `felt::FeltField::~~FeltField ()`

14.38.3 Member Function Documentation

14.38.3.1 `bool felt::FeltField::valid () const` `[inline]`

References `gridSize()`.

14.38.3.2 `int felt::FeltField::producer () const` `[inline]`

14.38.3.3 `int felt::FeltField::gridArea () const` `[inline]`

14.38.3.4 `boost::posix_time::ptime felt::FeltField::referenceTime () const`

14.38.3.5 `boost::posix_time::ptime felt::FeltField::validTime () const`

14.38.3.6 `int felt::FeltField::parameter () const` `[inline]`

14.38.3.7 `int felt::FeltField::dataType () const` `[inline]`

read the dataType, i.e. 1=analysis 2=interpolated/initialization 3=prognosis

14.38.3.8 `int felt::FeltField::verticalCoordinate () const` `[inline]`

14.38.3.9 `int felt::FeltField::level1 () const` `[inline]`

14.38.3.10 `int felt::FeltField::level2 () const` `[inline]`

14.38.3.11 `int felt::FeltField::gridType () const` `[inline]`

14.38.3.12 `void felt::FeltField::grid (std::vector< word > &out) const`

Read the grid from file.

14.38.3.13 `size_t felt::FeltField::gridSize () const`

Referenced by `valid()`.

14.38.3.14 `int felt::FeltField::scaleFactor () const`

14.38.3.15 `int felt::FeltField::xNum () const`

14.38.3.16 `int felt::FeltField::yNum () const`

14.38.3.17 `int felt::FeltField::miscField () const`

this field is described in the [felt](#) documentation as "word 19 in data part"

14.38.3.18 `int felt::FeltField::dataVersion () const`

Get data version if this is an eps parameter, otherwise 0

14.38.3.19 `FeltGridDefinitionPtr felt::FeltField::projectionInformation () const`

throws

14.38.3.20 `std::string felt::FeltField::information () const`

14.38.3.21 `std::string felt::FeltField::gridInformation () const`

14.38.3.22 `const Header& felt::FeltField::getHeader () const` `[inline]`

access [felt](#) index header

The documentation for this class was generated from the following file:

- `include/felt/FeltField.h`

14.39 felt::FeltFile Class Reference

```
#include <FeltFile.h>
```

Public Types

- typedef size_t [size_type](#)
- typedef boost::shared_ptr< [FeltField](#) > [FeltFieldPtr](#)
- typedef **std::vector**< [FeltFieldPtr](#) >::const_iterator [iterator](#)
- typedef [iterator](#) [const_iterator](#)

Public Member Functions

- [FeltFile](#) (const boost::filesystem::path &file)
- [~FeltFile](#) ()
- [size_type](#) [size](#) () const
- bool [empty](#) () const
- const boost::filesystem::path & [fileName](#) () const
- **std::string** [information](#) () const
- boost::posix_time::ptime [lastUpdateTime](#) () const
- boost::posix_time::ptime [referenceTime](#) () const
- boost::posix_time::ptime [firstTime](#) () const
- boost::posix_time::ptime [lastTime](#) () const
- [iterator](#) [begin](#) ()
- [iterator](#) [end](#) ()
- [const_iterator](#) [begin](#) () const
- [const_iterator](#) [end](#) () const
- const [FeltField](#) & [at](#) (size_t idx) const

*throws **std::out_of_range** if idx is too large.*

Static Public Member Functions

- static void [log](#) (const **std::string** &msg)
- static void [setLogStream](#) (**std::ostream** &o)
- static void [setLogging](#) (bool enableLogging)
- static bool [isLogging](#) ()

Friends

- class [FeltField](#)

14.39.1 Member Typedef Documentation

14.39.1.1 `typedef size_t felt::FeltFile::size_type`

14.39.1.2 `typedef boost::shared_ptr<FeltField> felt::FeltFile::FeltFieldPtr`

14.39.1.3 `typedef std::vector<FeltFieldPtr>::const_iterator felt::FeltFile::iterator`

14.39.1.4 `typedef iterator felt::FeltFile::const_iterator`

14.39.2 Constructor & Destructor Documentation

14.39.2.1 `felt::FeltFile::FeltFile (const boost::filesystem::path &file) [explicit]`

14.39.2.2 `felt::FeltFile::~~FeltFile ()`

14.39.3 Member Function Documentation

14.39.3.1 `size_type felt::FeltFile::size () const`

Referenced by `empty()`.

14.39.3.2 `bool felt::FeltFile::empty () const [inline]`

References `size()`.

14.39.3.3 `const boost::filesystem::path& felt::FeltFile::fileName () const [inline]`

14.39.3.4 `std::string felt::FeltFile::information () const`

14.39.3.5 `boost::posix_time::ptime felt::FeltFile::lastUpdateTime () const`

14.39.3.6 `boost::posix_time::ptime felt::FeltFile::referenceTime () const`

14.39.3.7 `boost::posix_time::ptime felt::FeltFile::firstTime () const`

14.39.3.8 `boost::posix_time::ptime felt::FeltFile::lastTime () const`

14.39.3.9 `iterator felt::FeltFile::begin ()`

14.39.3.10 `iterator felt::FeltFile::end ()`

14.39.3.11 `const_iterator felt::FeltFile::begin () const`

14.39.3.12 `const_iterator felt::FeltFile::end () const`

14.39.3.13 `const FeltField& felt::FeltFile::at (size_t idx) const`

throws `std::out_of_range` if `idx` is too large.

14.39.3.14 static void felt::FeltFile::log (const std::string & *msg*) [static]

14.39.3.15 static void felt::FeltFile::setLogStream (std::ostream & *o*) [static]

14.39.3.16 static void felt::FeltFile::setLogging (bool *enableLogging*) [static]

14.39.3.17 static bool felt::FeltFile::isLogging () [static]

14.39.4 Friends And Related Function Documentation

14.39.4.1 friend class FeltField [friend]

The documentation for this class was generated from the following file:

- include/felt/[FeltFile.h](#)

14.40 felt::FeltGridDefinition Class Reference

```
#include <FeltGridDefinition.h>
```

Public Types

- enum [Orientation](#) { [LeftUpperHorizontal](#) = 0, [LeftLowerHorizontal](#) = 64 }

Public Member Functions

- [FeltGridDefinition](#) (int gridType, int xNum, int yNum, int a, int b, int c, int d, const **std::vector**< short int > &extraData)
- virtual [~FeltGridDefinition](#) ()
- virtual **std::string** [projDefinition](#) () const
- virtual int [getXNumber](#) () const
- virtual int [getYNumber](#) () const
- virtual float [getXIncrement](#) () const
- virtual float [getYIncrement](#) () const
- virtual float [startLongitude](#) () const
- virtual float [startLatitude](#) () const
- virtual float [startX](#) () const
- virtual float [startY](#) () const
- virtual const boost::array< float, 6 > & [getGridParameters](#) () const
- [Orientation](#) [getScanMode](#) () const

14.40.1 Member Enumeration Documentation

14.40.1.1 enum felt::FeltGridDefinition::Orientation

Orientation describes the different ways that the values can be ordered in the grid. There are four possible dimensions: Left to Right or Right to Left Lower to Upper or Upper to Lower Horizontal scanning or Vertical scanning Regular or Alternating (i.e., every second row changes direction)

Enumerator:

LeftUpperHorizontal

LeftLowerHorizontal

14.40.2 Constructor & Destructor Documentation

14.40.2.1 felt::FeltGridDefinition::FeltGridDefinition (int gridType, int xNum, int yNum, int a, int b, int c, int d, const **std::vector**< short int > &extraData)

The parameters a, b, c, d are words 15 to 18 in the FELT header definition. These usually describe elements of the grid specification (variable meaning, depending on the grid specification used)

14.40.2.2 virtual felt::FeltGridDefinition::~~FeltGridDefinition () [virtual]

14.40.3 Member Function Documentation

14.40.3.1 virtual std::string felt::FeltGridDefinition::projDefinition () const [virtual]

14.40.3.2 virtual int felt::FeltGridDefinition::getXNumber () const [virtual]

14.40.3.3 virtual int felt::FeltGridDefinition::getYNumber () const [virtual]

14.40.3.4 virtual float felt::FeltGridDefinition::getXIncrement () const [virtual]

14.40.3.5 virtual float felt::FeltGridDefinition::getYIncrement () const [virtual]

14.40.3.6 virtual float felt::FeltGridDefinition::startLongitude () const [virtual]

14.40.3.7 virtual float felt::FeltGridDefinition::startLatitude () const [virtual]

14.40.3.8 virtual float felt::FeltGridDefinition::startX () const [virtual]

14.40.3.9 virtual float felt::FeltGridDefinition::startY () const [virtual]

14.40.3.10 virtual const boost::array<float, 6>& felt::FeltGridDefinition::getGridParameters () const [virtual]

14.40.3.11 Orientation felt::FeltGridDefinition::getScanMode () const

The documentation for this class was generated from the following file:

- include/felt/[FeltGridDefinition.h](#)

14.41 MetNoFelt::FeltParameters Class Reference

```
#include <FeltParameters.h>
```

Public Member Functions

- [FeltParameters](#) ()
- [FeltParameters](#) (std::string filename)
- [FeltParameters](#) (const std::vector< std::string > &feltParams)
- virtual [~FeltParameters](#) ()
- const boost::array< short, 16 > & [getParameters](#) (const std::string &)
- const std::string & [getParameterName](#) (const boost::array< short, 16 > &)
- std::string [getParameterDatatype](#) (const std::string ¶meterName) const
- double [getParameterFillValue](#) (const std::string ¶meterName) const

Static Public Member Functions

- static const std::string & [DEFAULT_CONFIG](#) ()

14.41.1 Constructor & Destructor Documentation

14.41.1.1 MetNoFelt::FeltParameters::FeltParameters ()

14.41.1.2 MetNoFelt::FeltParameters::FeltParameters (std::string filename) [explicit]

initialize all known [felt](#) parameters from a diana-setup file

Parameters:

filename diana setup file

14.41.1.3 MetNoFelt::FeltParameters::FeltParameters (const std::vector< std::string > &feltParams) [explicit]

initialize parameters from a list of parameters in diana format, e.g. 17,2,1000:prod=74

14.41.1.4 `virtual MetNoFelt::FeltParameters::~~FeltParameters ()` [virtual]

14.41.2 Member Function Documentation

14.41.2.1 `const boost::array<short, 16>& MetNoFelt::FeltParameters::getParameters (const std::string &)`

14.41.2.2 `const std::string& MetNoFelt::FeltParameters::getParameterName (const boost::array< short, 16 > &)`

14.41.2.3 `std::string MetNoFelt::FeltParameters::getParameterDatatype (const std::string & parameterName) const`

14.41.2.4 `double MetNoFelt::FeltParameters::getParameterFillValue (const std::string & parameterName) const`

14.41.2.5 `static const std::string& MetNoFelt::FeltParameters::DEFAULT_CONFIG ()`
[inline, static]

The documentation for this class was generated from the following file:

- include/fimex/[FeltParameters.h](#)

14.42 MetNoFimex::FimexTime Class Reference

```
#include <TimeUnit.h>
```

Public Member Functions

- bool `operator==` (const [FimexTime](#) &rhs) const
compare two fimexTimes
- bool `operator!=` (const [FimexTime](#) &rhs) const
compare two fimexTimes
- bool `operator>` (const [FimexTime](#) &rhs) const
compare two fimexTimes
- bool `operator<` (const [FimexTime](#) &rhs) const
compare two fimexTimes
- bool `operator>=` (const [FimexTime](#) &rhs) const
compare two fimexTimes
- bool `operator<=` (const [FimexTime](#) &rhs) const
compare two fimexTimes

Public Attributes

- unsigned short [msecond](#)
millisecond
- char [second](#)
second (0-59)
- char [minute](#)
minute (0-59)
- char [hour](#)
hour (0-23)
- char [mday](#)
day of month (1-31)
- char [month](#)
month (1-12)
- unsigned short [year](#)
year (2008 as of writing)

14.42.1 Detailed Description

time representation and some overloaded operators

Warning:

: the implementor needs to make sure, that all values are given correctly, i.e. seconds between 0 and 59

14.42.2 Member Function Documentation

14.42.2.1 `bool MetNoFimex::FimexTime::operator==(const FimexTime & rhs) const`

compare two fimexTimes

14.42.2.2 `bool MetNoFimex::FimexTime::operator!=(const FimexTime & rhs) const` [inline]

compare two fimexTimes

14.42.2.3 `bool MetNoFimex::FimexTime::operator>(const FimexTime & rhs) const` [inline]

compare two fimexTimes

References toLong().

14.42.2.4 `bool MetNoFimex::FimexTime::operator<(const FimexTime & rhs) const` [inline]

compare two fimexTimes

14.42.2.5 `bool MetNoFimex::FimexTime::operator>=(const FimexTime & rhs) const` [inline]

compare two fimexTimes

14.42.2.6 `bool MetNoFimex::FimexTime::operator<=(const FimexTime & rhs) const` [inline]

compare two fimexTimes

14.42.3 Member Data Documentation

14.42.3.1 `unsigned short MetNoFimex::FimexTime::msecond`

millisecond

14.42.3.2 `char MetNoFimex::FimexTime::second`

second (0-59)

14.42.3.3 char MetNoFimex::FimexTime::minute

minute (0-59)

14.42.3.4 char MetNoFimex::FimexTime::hour

hour (0-23)

14.42.3.5 char MetNoFimex::FimexTime::mday

day of month (1-31)

14.42.3.6 char MetNoFimex::FimexTime::month

month (1-12)

14.42.3.7 unsigned short MetNoFimex::FimexTime::year

year (2008 as of writing)

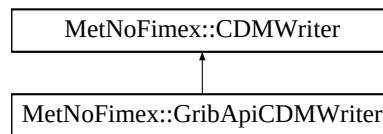
The documentation for this class was generated from the following file:

- include/fimex/[TimeUnit.h](#)

14.43 MetNoFimex::GribApiCDMWriter Class Reference

```
#include <GribApiCDMWriter.h>
```

Inheritance diagram for MetNoFimex::GribApiCDMWriter::



Public Member Functions

- [GribApiCDMWriter](#) (const boost::shared_ptr< [CDMReader](#) > [cdmReader](#), const std::string &[outputFile](#), const int [gribVersion](#), const std::string &[configFile](#))
- virtual [~GribApiCDMWriter](#) ()

14.43.1 Constructor & Destructor Documentation

14.43.1.1 MetNoFimex::GribApiCDMWriter::GribApiCDMWriter (const boost::shared_ptr< CDMReader > *cdmReader*, const std::string & *outputFile*, const int *gribVersion*, const std::string & *configFile*)

14.43.1.2 virtual MetNoFimex::GribApiCDMWriter::~~GribApiCDMWriter () [virtual]

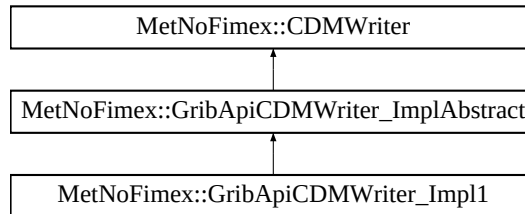
The documentation for this class was generated from the following file:

- include/fimex/[GribApiCDMWriter.h](#)

14.44 MetNoFimex::GribApiCDMWriter_Impl1 Class Reference

```
#include <GribApiCDMWriter_Impl1.h>
```

Inheritance diagram for MetNoFimex::GribApiCDMWriter_Impl1::



Public Member Functions

- [GribApiCDMWriter_Impl1](#) (const boost::shared_ptr< [CDMReader](#) > &cdmReader, const std::string &outputFile, const std::string &configFile)
- virtual [~GribApiCDMWriter_Impl1](#) ()
- virtual void [setParameter](#) (const std::string &varName, const [FimexTime](#) &fTime, double levelValue) throw (CDMException)
- virtual void [setProjection](#) (const std::string &varName) throw (CDMException)
- virtual void [setLevel](#) (const std::string &varName, double levelValue)
- virtual boost::shared_ptr< [Data](#) > [handleTypeScaleAndMissingData](#) (const std::string &varName, const [FimexTime](#) &fTime, double levelValue, boost::shared_ptr< [Data](#) > inData)

14.44.1 Detailed Description

Implementaionn of a writer using GribApi for grib1

14.44.2 Constructor & Destructor Documentation

14.44.2.1 MetNoFimex::GribApiCDMWriter_Impl1::GribApiCDMWriter_Impl1 (const boost::shared_ptr< [CDMReader](#) > & cdmReader, const std::string & outputFile, const std::string & configFile)

14.44.2.2 virtual MetNoFimex::GribApiCDMWriter_Impl1::~~GribApiCDMWriter_Impl1 ()
[virtual]

14.44.3 Member Function Documentation

14.44.3.1 virtual void MetNoFimex::GribApiCDMWriter_Impl1::setParameter (const std::string & varName, const [FimexTime](#) & fTime, double levelValue) throw (CDMException)
[virtual]

Implements [MetNoFimex::GribApiCDMWriter_ImplAbstract](#).

14.44.3.2 virtual void MetNoFimex::GribApiCDMWriter_Impl1::setProjection (const std::string & varName) throw (CDMException) [virtual]

set the projection parameters, throw an exception if none are available

Parameters:

varName

Exceptions:

[*CDMException*](#) if parameters cannot be set

Implements [MetNoFimex::GribApiCDMWriter_ImplAbstract](#).

14.44.3.3 virtual void MetNoFimex::GribApiCDMWriter_Impl1::setLevel (const std::string & varName, double levelValue) [virtual]

Implements [MetNoFimex::GribApiCDMWriter_ImplAbstract](#).

14.44.3.4 virtual boost::shared_ptr<Data> MetNoFimex::GribApiCDMWriter_Impl1::handleTypeScaleAndMissingData (const std::string & varName, const FimexTime & fTime, double levelValue, boost::shared_ptr< Data > inData) [virtual]

add the missing value to the gribHandle, rescale the data if needed and change the datatype if needed, change the missingValue of the data if need

Returns:

modified data

Implements [MetNoFimex::GribApiCDMWriter_ImplAbstract](#).

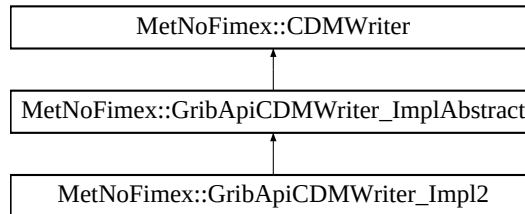
The documentation for this class was generated from the following file:

- [include/fimex/GribApiCDMWriter_Impl1.h](#)

14.45 MetNoFimex::GribApiCDMWriter_Impl2 Class Reference

```
#include <GribApiCDMWriter_Impl2.h>
```

Inheritance diagram for MetNoFimex::GribApiCDMWriter_Impl2::



Public Member Functions

- [GribApiCDMWriter_Impl2](#) (const boost::shared_ptr< [CDMReader](#) > &cdmReader, const std::string &outputFile, const std::string &configFile)
- virtual [~GribApiCDMWriter_Impl2](#) ()
- virtual void [setParameter](#) (const std::string &varName, const [FimexTime](#) &fTime, double levelValue) throw (CDMException)
- virtual void [setProjection](#) (const std::string &varName) throw (CDMException)
- virtual void [setLevel](#) (const std::string &varName, double levelValue)
- virtual boost::shared_ptr< [Data](#) > [handleTypeScaleAndMissingData](#) (const std::string &varName, const [FimexTime](#) &fTime, double levelValue, boost::shared_ptr< [Data](#) > inData)

14.45.1 Detailed Description

Implementaionn of a writer using GribApi for grib2

14.45.2 Constructor & Destructor Documentation

14.45.2.1 MetNoFimex::GribApiCDMWriter_Impl2::GribApiCDMWriter_Impl2 (const boost::shared_ptr< CDMReader > & cdmReader, const std::string & outputFile, const std::string & configFile)

14.45.2.2 virtual MetNoFimex::GribApiCDMWriter_Impl2::~~GribApiCDMWriter_Impl2 ()
[virtual]

14.45.3 Member Function Documentation

14.45.3.1 virtual void MetNoFimex::GribApiCDMWriter_Impl2::setParameter (const std::string & varName, const FimexTime & fTime, double levelValue) throw (CDMException)
[virtual]

Implements [MetNoFimex::GribApiCDMWriter_ImplAbstract](#).

14.45.3.2 virtual void MetNoFimex::GribApiCDMWriter_Impl2::setProjection (const std::string & *varName*) throw (CDMException) [virtual]

set the projection parameters, throw an exception if none are available

Parameters:

varName

Exceptions:

[*CDMException*](#) if parameters cannot be set

Implements [MetNoFimex::GribApiCDMWriter_ImplAbstract](#).

14.45.3.3 virtual void MetNoFimex::GribApiCDMWriter_Impl2::setLevel (const std::string & *varName*, double *levelValue*) [virtual]

Implements [MetNoFimex::GribApiCDMWriter_ImplAbstract](#).

14.45.3.4 virtual boost::shared_ptr<Data> MetNoFimex::GribApiCDMWriter_Impl2::handleTypeScaleAndMissingData (const std::string & *varName*, const FimexTime & *fTime*, double *levelValue*, boost::shared_ptr< Data > *inData*) [virtual]

add the missing value to the gribHandle, rescale the data if needed and change the datatype if needed, change the missingValue of the data if need

Returns:

modified data

Implements [MetNoFimex::GribApiCDMWriter_ImplAbstract](#).

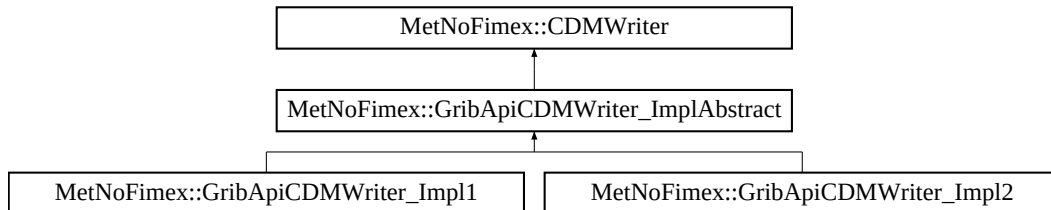
The documentation for this class was generated from the following file:

- [include/fimex/GribApiCDMWriter_Impl2.h](#)

14.46 MetNoFimex::GribApiCDMWriter_ImplAbstract Class Reference

```
#include <GribApiCDMWriter_ImplAbstract.h>
```

Inheritance diagram for MetNoFimex::GribApiCDMWriter_ImplAbstract:



Public Member Functions

- [GribApiCDMWriter_ImplAbstract](#) (int [gribVersion](#), const boost::shared_ptr< [CDMReader](#) > &[cdmReader](#), const **std::string** &[outputFile](#), const **std::string** &[configFile](#))
- virtual [~GribApiCDMWriter_ImplAbstract](#) ()
- void [run](#) () throw (CDMException)
actually write the data

Protected Member Functions

- virtual void [setGlobalAttributes](#) ()
- virtual void [setData](#) (const boost::shared_ptr< [Data](#) > &[data](#))
- virtual void [setProjection](#) (const **std::string** &[varName](#))=0 throw (CDMException)
- virtual void [setParameter](#) (const **std::string** &[varName](#), const [FimexTime](#) &[fTime](#), double levelValue)=0 throw (CDMException)
- virtual void [setTime](#) (const **std::string** &[varName](#), const [FimexTime](#) &[fTime](#))
- virtual void [setLevel](#) (const **std::string** &[varName](#), double levelValue)=0
- virtual **std::vector**< double > [getLevels](#) (const **std::string** &[varName](#)) throw (CDMException)
- virtual **std::vector**< [FimexTime](#) > [getTimes](#) (const **std::string** &[varName](#)) throw (CDMException)
- virtual boost::shared_ptr< [Data](#) > [handleTypeScaleAndMissingData](#) (const **std::string** &[varName](#), const [FimexTime](#) &[fTime](#), double levelValue, boost::shared_ptr< [Data](#) > [inData](#))=0
- virtual void [writeGribHandleToFile](#) ()
- [xmlNode](#) * [getNodePtr](#) (const **std::string** &[varName](#), const [FimexTime](#) &[fTime](#), double levelValue) throw (CDMException)

Protected Attributes

- int [gribVersion](#)
- const **std::string** [configFile](#)
- const boost::shared_ptr< [XMLDoc](#) > [xmlConfig](#)
- boost::shared_ptr< [grib_handle](#) > [gribHandle](#)
- [LoggerPtr](#) [logger](#)

14.46.1 Constructor & Destructor Documentation

14.46.1.1 `MetNoFimex::GribApiCDMWriter_ImplAbstract::GribApiCDMWriter_ImplAbstract (int gribVersion, const boost::shared_ptr< CDMReader > & cdmReader, const std::string & outputFile, const std::string & configFile)`

Constructor of the general writer. It should be called during construction of derived classes.

remember to call run to actually do something

14.46.1.2 `virtual MetNoFimex::GribApiCDMWriter_ImplAbstract::~~GribApiCDMWriter_ImplAbstract () [virtual]`

14.46.2 Member Function Documentation

14.46.2.1 `void MetNoFimex::GribApiCDMWriter_ImplAbstract::run () throw (CDMException)`

actually write the data

The run function has be to called after construction the object to actually fetch and write the data.

14.46.2.2 `virtual void MetNoFimex::GribApiCDMWriter_ImplAbstract::setGlobalAttributes () [protected, virtual]`

add the global attributes from the config to the default grib-handle

14.46.2.3 `virtual void MetNoFimex::GribApiCDMWriter_ImplAbstract::setData (const boost::shared_ptr< Data > & data) [protected, virtual]`

14.46.2.4 `virtual void MetNoFimex::GribApiCDMWriter_ImplAbstract::setProjection (const std::string & varName) throw (CDMException) [protected, pure virtual]`

set the projection parameters, throw an exception if none are available

Parameters:

varName

Exceptions:

CDMException if parameters cannot be set

Implemented in [MetNoFimex::GribApiCDMWriter_Impl1](#), and [MetNoFimex::GribApiCDMWriter_Impl2](#).

14.46.2.5 `virtual void MetNoFimex::GribApiCDMWriter_ImplAbstract::setParameter (const std::string & varName, const FimexTime & fTime, double levelValue) throw (CDMException) [protected, pure virtual]`

Implemented in [MetNoFimex::GribApiCDMWriter_Impl1](#), and [MetNoFimex::GribApiCDMWriter_Impl2](#).

14.46.2.6 `virtual void MetNoFimex::GribApiCDMWriter_ImplAbstract::setTime (const std::string & varName, const FimexTime & fTime)` [protected, virtual]

14.46.2.7 `virtual void MetNoFimex::GribApiCDMWriter_ImplAbstract::setLevel (const std::string & varName, double levelValue)` [protected, pure virtual]

Implemented in [MetNoFimex::GribApiCDMWriter_Impl1](#), and [MetNoFimex::GribApiCDMWriter_Impl2](#).

14.46.2.8 `virtual std::vector<double> MetNoFimex::GribApiCDMWriter_ImplAbstract::getLevels (const std::string & varName)` throw (CDMException) [protected, virtual]

get the levels from the cdm scaled to values used in grib (units/scale-factor) assign at least 1 level, give it a default value if none is found in the cdm

14.46.2.9 `virtual std::vector<FimexTime> MetNoFimex::GribApiCDMWriter_ImplAbstract::getTimes (const std::string & varName)` throw (CDMException) [protected, virtual]

get the times from the cdm as [FimexTime](#) (including unit) assign at least 1 time, give it a default value if none is found in the cdm

14.46.2.10 `virtual boost::shared_ptr<Data> MetNoFimex::GribApiCDMWriter_ImplAbstract::handleTypeScaleAndMissingData (const std::string & varName, const FimexTime & fTime, double levelValue, boost::shared_ptr<Data> inData)` [protected, pure virtual]

add the missing value to the gribHandle, rescale the data if needed and change the datatype if needed, change the missingValue of the data if need

Returns:

modified data

Implemented in [MetNoFimex::GribApiCDMWriter_Impl1](#), and [MetNoFimex::GribApiCDMWriter_Impl2](#).

14.46.2.11 `virtual void MetNoFimex::GribApiCDMWriter_ImplAbstract::writeGribHandleToFile ()` [protected, virtual]

14.46.2.12 `xmlNode* MetNoFimex::GribApiCDMWriter_ImplAbstract::getNodePtr (const std::string & varName, const FimexTime & fTime, double levelValue)` throw (CDMException) [protected]

get the node belonging to varName, level and time from the config file

Parameters:

varName name of the variable

fTime current time

level current level

14.46.3 Member Data Documentation

14.46.3.1 `int MetNoFimex::GribApiCDMWriter_ImplAbstract::gribVersion` [protected]

14.46.3.2 `const std::string MetNoFimex::GribApiCDMWriter_ImplAbstract::configFile`
[protected]

14.46.3.3 `const boost::shared_ptr<XMLDoc> MetNoFimex::GribApiCDMWriter_ImplAbstract::xmlConfig` [protected]

14.46.3.4 `boost::shared_ptr<grib_handle> MetNoFimex::GribApiCDMWriter_ImplAbstract::gribHandle` [protected]

14.46.3.5 `LoggerPtr MetNoFimex::GribApiCDMWriter_ImplAbstract::logger` [protected]

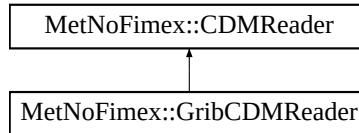
The documentation for this class was generated from the following file:

- [include/fimex/GribApiCDMWriter_ImplAbstract.h](#)

14.47 MetNoFimex::GribCDMReader Class Reference

```
#include <GribCDMReader.h>
```

Inheritance diagram for MetNoFimex::GribCDMReader::



Public Member Functions

- [GribCDMReader](#) (const **std::vector**< **std::string** > &fileNames, const **std::string** &configFile)
- virtual [~GribCDMReader](#) ()
- virtual boost::shared_ptr< [Data](#) > [getDataSlice](#) (const **std::string** &varName, size_t unLimDimPos) throw (CDMException)
data-reading function to be called from the [CDMWriter](#)

14.47.1 Constructor & Destructor Documentation

14.47.1.1 MetNoFimex::GribCDMReader::GribCDMReader (const **std::vector**< **std::string** > &fileNames, const **std::string** &configFile)

14.47.1.2 virtual MetNoFimex::GribCDMReader::~~GribCDMReader () [virtual]

14.47.2 Member Function Documentation

14.47.2.1 virtual boost::shared_ptr< [Data](#) > MetNoFimex::GribCDMReader::getDataSlice (const **std::string** &varName, size_t unLimDimPos) throw (CDMException) [virtual]

data-reading function to be called from the [CDMWriter](#)

This methods needs to be implemented by the [CDMReader](#). It should provide the data for each variable, either by reading from disk, converting from another [CDMReader](#) or reading from an in-memory data-section.

This function should retrieve the whole data for a dataset without unlimited dimension if the unLimDimPos == 0.

Parameters:

varName name of the variable to read

unLimDimPos (optional) if the variable contains a unlimited dimension (max one allowed) an slice of this position is returned

Implements [MetNoFimex::CDMReader](#).

The documentation for this class was generated from the following file:

- include/fimex/[GribCDMReader.h](#)

14.48 MetNoFimex::GribFileIndex Class Reference

```
#include <GribFileIndex.h>
```

Public Member Functions

- [GribFileIndex](#) ()
- [GribFileIndex](#) (boost::filesystem::path gribFilePath, bool ignoreExistingXml=false)
- virtual [~GribFileIndex](#) ()
- const **std::vector**< [GribFileMessage](#) > & [listMessages](#) () const
- const **std::string** & [getUrl](#) () const

14.48.1 Constructor & Destructor Documentation

14.48.1.1 MetNoFimex::GribFileIndex::GribFileIndex ()

14.48.1.2 MetNoFimex::GribFileIndex::GribFileIndex (boost::filesystem::path *gribFilePath*, bool *ignoreExistingXml* = false)

Initialize the gribFileIndex for the gribFile gribFilePath. If ignoreExistingXml = false, searches for existing indexes in

- file.grbml
- ENV{GRIB_FILE_INDEX}/file.grbml

Otherwise, it parses the grib-file and creates a index.

Performance for getting an index of a 150MB grib-file with some 10s of messages:

- remote NFS file, first time: 16s
- file completely in memory: 1.1s
- xml-file: 0.1s

14.48.1.3 virtual MetNoFimex::GribFileIndex::~~GribFileIndex () [virtual]

14.48.2 Member Function Documentation

14.48.2.1 const std::vector<GribFileMessage>& MetNoFimex::GribFileIndex::listMessages () const [inline]

14.48.2.2 const std::string& MetNoFimex::GribFileIndex::getUrl () const [inline]

The documentation for this class was generated from the following file:

- include/fimex/[GribFileIndex.h](#)

14.49 MetNoFimex::GribFileMessage Class Reference

```
#include <GribFileIndex.h>
```

Public Member Functions

- [GribFileMessage](#) ()
- [GribFileMessage](#) (boost::shared_ptr< grib_handle > gh, const **std::string** &fileURL, long filePos, long msgPos)
- [GribFileMessage](#) (boost::shared_ptr< [XMLDoc](#) >, **std::string** nsPrefix, [xmlNodePtr](#) node)
- [~GribFileMessage](#) ()
- bool [isValid](#) () const
test if this is a proper [GribFileMessage](#) or just the default constructor
- **std::string** [toString](#) () const
give a xml-string representation
- const long [getEdition](#) () const
accessors
- const **std::string** & [getFileURL](#) () const
- const size_t [getFilePosition](#) () const
- const size_t [getMessageNumber](#) () const
messages number within a multi-message
- const **std::string** & [getName](#) () const
- const **std::string** & [getShortName](#) () const
- boost::posix_time::ptime [getDateTime](#) () const
- long [getLevelNumber](#) () const
- long [getLevelType](#) () const
- const **std::vector**< long > & [getParameterIds](#) () const
- const **std::string** & [getTypeOfGrid](#) () const
- const [GridDefinition](#) & [getGridDefinition](#) () const

14.49.1 Constructor & Destructor Documentation

14.49.1.1 MetNoFimex::GribFileMessage::GribFileMessage ()

14.49.1.2 MetNoFimex::GribFileMessage::GribFileMessage (boost::shared_ptr< grib_handle > gh, const **std::string** &fileURL, long filePos, long msgPos)

14.49.1.3 MetNoFimex::GribFileMessage::GribFileMessage (boost::shared_ptr< [XMLDoc](#) >, **std::string** nsPrefix, [xmlNodePtr](#) node)

14.49.1.4 MetNoFimex::GribFileMessage::~~GribFileMessage ()

14.49.2 Member Function Documentation

14.49.2.1 bool MetNoFimex::GribFileMessage::isValid () const [inline]

test if this is a proper [GribFileMessage](#) or just the default constructor

14.49.2.2 std::string MetNoFimex::GribFileMessage::toString () const

give a xml-string representation

14.49.2.3 const long MetNoFimex::GribFileMessage::getEdition () const

accessors

Referenced by MetNoFimex::GribFileMessageEqualLevelTime::operator().

14.49.2.4 const std::string& MetNoFimex::GribFileMessage::getFileURL () const**14.49.2.5 const size_t MetNoFimex::GribFileMessage::getFilePosition () const****14.49.2.6 const size_t MetNoFimex::GribFileMessage::getMessageNumber () const**

messages number within a multi-message

14.49.2.7 const std::string& MetNoFimex::GribFileMessage::getName () const**14.49.2.8 const std::string& MetNoFimex::GribFileMessage::getShortName () const****14.49.2.9 boost::posix_time::ptime MetNoFimex::GribFileMessage::getDateTime () const**

Referenced by MetNoFimex::GribFileMessageEqualLevelTime::operator(), and MetNoFimex::GribFileMessageEqualTime::operator().

14.49.2.10 long MetNoFimex::GribFileMessage::getLevelNumber () const

Referenced by MetNoFimex::GribFileMessageEqualLevelTime::operator().

14.49.2.11 long MetNoFimex::GribFileMessage::getLevelType () const

Referenced by MetNoFimex::GribFileMessageEqualLevelTime::operator().

14.49.2.12 const std::vector<long>& MetNoFimex::GribFileMessage::getParameterIds () const

Get the parameter ids as list with the following meanings:

- ed1: indicatorOfParameter, gribTablesVersionNo, identificationOfOriginatingGeneratingCentre;
- ed2: parameterNumber, paramterCategory, discipline

14.49.2.13 const std::string& MetNoFimex::GribFileMessage::getTypeOfGrid () const**14.49.2.14 const GridDefinition& MetNoFimex::GribFileMessage::getGridDefinition () const**

The documentation for this class was generated from the following file:

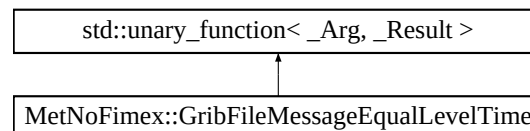
- `include/fimex/GribFileIndex.h`

14.50 MetNoFimex::GribFileMessageEqualLevelTime Class Reference

Functor to find messages with equal level and time.

```
#include <GribFileIndex.h>
```

Inheritance diagram for MetNoFimex::GribFileMessageEqualLevelTime::



Public Member Functions

- [GribFileMessageEqualLevelTime](#) (long *edition*, long *levelType*, long *levelNo*, boost::posix_time::ptime *time*)
- [~GribFileMessageEqualLevelTime](#) ()
- bool [operator\(\)](#) (const [GribFileMessage](#) &*gfm*)

14.50.1 Detailed Description

Functor to find messages with equal level and time.

14.50.2 Constructor & Destructor Documentation

14.50.2.1 MetNoFimex::GribFileMessageEqualLevelTime::GribFileMessageEqualLevelTime (long *edition*, long *levelType*, long *levelNo*, boost::posix_time::ptime *time*) [inline]

14.50.2.2 MetNoFimex::GribFileMessageEqualLevelTime::~~GribFileMessageEqualLevelTime () [inline]

14.50.3 Member Function Documentation

14.50.3.1 bool MetNoFimex::GribFileMessageEqualLevelTime::operator() (const [GribFileMessage](#) &*gfm*) [inline]

References MetNoFimex::GribFileMessage::getDateTime(), MetNoFimex::GribFileMessage::getEdition(), MetNoFimex::GribFileMessage::getLevelNumber(), and MetNoFimex::GribFileMessage::getLevelType().

The documentation for this class was generated from the following file:

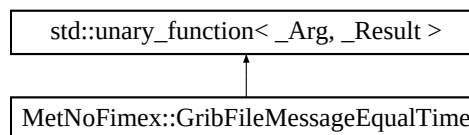
- include/fimex/[GribFileIndex.h](#)

14.51 MetNoFimex::GribFileMessageEqualTime Class Reference

Functor to find Messages with equal time.

```
#include <GribFileIndex.h>
```

Inheritance diagram for MetNoFimex::GribFileMessageEqualTime::



Public Member Functions

- [GribFileMessageEqualTime](#) (boost::posix_time::ptime time)
- [~GribFileMessageEqualTime](#) ()
- bool [operator\(\)](#) (const [GribFileMessage](#) &gfm)

14.51.1 Detailed Description

Functor to find Messages with equal time.

14.51.2 Constructor & Destructor Documentation

14.51.2.1 MetNoFimex::GribFileMessageEqualTime::GribFileMessageEqualTime (boost::posix_time::ptime *time*) [inline]

14.51.2.2 MetNoFimex::GribFileMessageEqualTime::~~GribFileMessageEqualTime () [inline]

14.51.3 Member Function Documentation

14.51.3.1 bool MetNoFimex::GribFileMessageEqualTime::operator() (const [GribFileMessage](#) &*gfm*) [inline]

References [MetNoFimex::GribFileMessage::getDateTime\(\)](#).

The documentation for this class was generated from the following file:

- include/fimex/[GribFileIndex.h](#)

14.52 MetNoFimex::GridDefinition Class Reference

```
#include <GridDefinition.h>
```

Public Types

- enum [OrientationFlags](#) { [ScanStartRight](#) = binary<010000000>::value, [ScanStartBottom](#) = binary<001000000>::value, [ScanIsVertical](#) = binary<000100000>::value, [ScanIsAlternating](#) = binary<000010000>::value }
- enum [Orientation](#) {
[LeftUpperHorizontal](#) = binary<000000000>::value, [RightUpperHorizontal](#) = binary<010000000>::value, [LeftLowerHorizontal](#) = binary<001000000>::value, [RightLowerHorizontal](#) = binary<011000000>::value,
[LeftUpperVertical](#) = binary<000100000>::value, [RightUpperVertical](#) = binary<010100000>::value, [LeftLowerVertical](#) = binary<001100000>::value, [RightLowerVertical](#) = binary<011100000>::value,
[LeftUpperHorizontalAlternating](#) = binary<000010000>::value, [RightUpperHorizontalAlternating](#) = binary<010010000>::value, [LeftLowerHorizontalAlternating](#) = binary<001010000>::value, [RightLowerHorizontalAlternating](#) = binary<011010000>::value,
[LeftUpperVerticalAlternating](#) = binary<000110000>::value, [RightUpperVerticalAlternating](#) = binary<010110000>::value, [LeftLowerVerticalAlternating](#) = binary<001110000>::value, [RightLowerVerticalAlternating](#) = binary<011110000>::value }

Public Member Functions

- [GridDefinition](#) ()
- [GridDefinition](#) (**std::string** projDefinition, size_t xSize, size_t ySize, double xIncr, double yIncr, double xStart, double yStart, [Orientation](#) orient)
- virtual [~GridDefinition](#) ()
- virtual **std::string** [getProjDefinition](#) () const
return a proj4 string
- virtual void [setProjDefinition](#) (**std::string** proj)
- virtual size_t [getXSize](#) () const
number of points in x or longitude direction
- virtual void [setXSize](#) (size_t xSize)
- virtual size_t [getYSize](#) () const
number of points in y or latitude direction
- virtual void [setYSize](#) (size_t ySize)
- virtual double [getXIncrement](#) () const
x or longitude increment in m or degree
- virtual void [setXIncrement](#) (double xIncr)
- virtual double [getYIncrement](#) () const
y or latitude increment in m or degree
- virtual void [setYIncrement](#) (double yIncr)

- virtual double [getXStart](#) () const
x or longitude start in m or degree
- virtual void [setXStart](#) (double startX)
- virtual double [getYStart](#) () const
y or latitude start in m or degree
- virtual void [setYStart](#) (double startY)
- virtual [Orientation](#) [getScanMode](#) () const
- virtual void [setScanMode](#) ([Orientation](#) orient)
- virtual bool [comparableTo](#) (const [GridDefinition](#) &rhs, double delta=0.) const

14.52.1 Member Enumeration Documentation

14.52.1.1 enum MetNoFimex::GridDefinition::OrientationFlags

Use these flags to build an Orientation. Left, Upper, Horizontal and not Alternating are defaults and don't require flags.

Enumerator:

ScanStartRight
ScanStartBottom
ScanIsVertical
ScanIsAlternating change direction between succeeding rows (horizontal) or columns (vertical)

14.52.1.2 enum MetNoFimex::GridDefinition::Orientation

Enumerator:

LeftUpperHorizontal
RightUpperHorizontal
LeftLowerHorizontal
RightLowerHorizontal
LeftUpperVertical
RightUpperVertical
LeftLowerVertical
RightLowerVertical
LeftUpperHorizontalAlternating
RightUpperHorizontalAlternating
LeftLowerHorizontalAlternating
RightLowerHorizontalAlternating
LeftUpperVerticalAlternating
RightUpperVerticalAlternating
LeftLowerVerticalAlternating
RightLowerVerticalAlternating

14.52.2 Constructor & Destructor Documentation

14.52.2.1 MetNoFimex::GridDefinition::GridDefinition ()

14.52.2.2 MetNoFimex::GridDefinition::GridDefinition (std::string *projDefinition*, size_t *xSize*, size_t *ySize*, double *xIncr*, double *yIncr*, double *xStart*, double *yStart*, Orientation *orient*)

14.52.2.3 virtual MetNoFimex::GridDefinition::~~GridDefinition () [virtual]

14.52.3 Member Function Documentation

14.52.3.1 virtual std::string MetNoFimex::GridDefinition::getProjDefinition () const [virtual]

return a proj4 string

14.52.3.2 virtual void MetNoFimex::GridDefinition::setProjDefinition (std::string *proj*) [virtual]

14.52.3.3 virtual size_t MetNoFimex::GridDefinition::getXSize () const [virtual]

number of points in x or longitude direction

14.52.3.4 virtual void MetNoFimex::GridDefinition::setXSize (size_t *xSize*) [virtual]

14.52.3.5 virtual size_t MetNoFimex::GridDefinition::getYSize () const [virtual]

number of points in y or latitude direction

14.52.3.6 virtual void MetNoFimex::GridDefinition::setYSize (size_t *ySize*) [virtual]

14.52.3.7 virtual double MetNoFimex::GridDefinition::getXIncrement () const [virtual]

x or longitude increment in m or degree

14.52.3.8 virtual void MetNoFimex::GridDefinition::setXIncrement (double *xIncr*) [virtual]

14.52.3.9 virtual double MetNoFimex::GridDefinition::getYIncrement () const [virtual]

y or latitude increment in m or degree

14.52.3.10 virtual void MetNoFimex::GridDefinition::setYIncrement (double *yIncr*) [virtual]

14.52.3.11 virtual double MetNoFimex::GridDefinition::getXStart () const [virtual]

x or longitude start in m or degree

14.52.3.12 `virtual void MetNoFimex::GridDefinition::setXStart (double startX)` [virtual]

14.52.3.13 `virtual double MetNoFimex::GridDefinition::getYStart () const` [virtual]

y or latitude start in m or degree

14.52.3.14 `virtual void MetNoFimex::GridDefinition::setYStart (double startY)` [virtual]

14.52.3.15 `virtual Orientation MetNoFimex::GridDefinition::getScanMode () const` [virtual]

14.52.3.16 `virtual void MetNoFimex::GridDefinition::setScanMode (Orientation orient)`
[virtual]

14.52.3.17 `virtual bool MetNoFimex::GridDefinition::comparableTo (const GridDefinition & rhs,
double delta = 0.) const` [virtual]

Compare two GridDefinitions. They are comparable if they have

- same size(XY)
- same incr(XY) within the delta
- same start(XY) within the delta

Parameters:

rhs the other gridDefinition

delta the relative delta to compare to (a == 0) ? (abs(b) <= delta) : abs((b-a)/a) <= delta

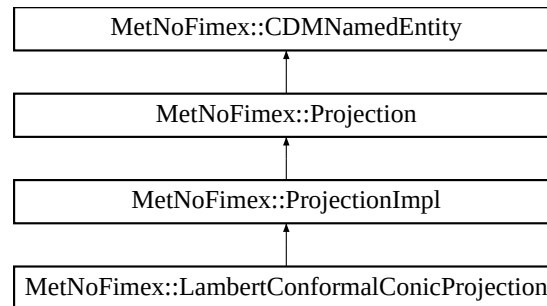
The documentation for this class was generated from the following file:

- include/fimex/[GridDefinition.h](#)

14.53 MetNoFimex::LambertConformalConicProjection Class Reference

```
#include <LambertConformalConicProjection.h>
```

Inheritance diagram for MetNoFimex::LambertConformalConicProjection::



Public Member Functions

- [LambertConformalConicProjection](#) ()
- virtual [~LambertConformalConicProjection](#) ()

Static Public Member Functions

- static bool [acceptsProj4](#) (const **std::string** &proj4Str)
- static **std::vector**< [CDMAttribute](#) > [parametersFromProj4](#) (const **std::string** &proj4)

Protected Member Functions

- virtual **std::ostream** & [getProj4ProjectionPart](#) (**std::ostream** &oproj) const

14.53.1 Constructor & Destructor Documentation

14.53.1.1 `MetNoFimex::LambertConformalConicProjection::LambertConformalConicProjection()`

14.53.1.2 `virtual MetNoFimex::LambertConformalConicProjection::~~LambertConformalConicProjection()` `[inline, virtual]`

14.53.2 Member Function Documentation

14.53.2.1 `static bool MetNoFimex::LambertConformalConicProjection::acceptsProj4(const std::string &proj4Str)` `[static]`

14.53.2.2 `static std::vector<CDMAAttribute> MetNoFimex::LambertConformalConicProjection::parametersFromProj4(const std::string &proj4)` `[static]`

14.53.2.3 `virtual std::ostream& MetNoFimex::LambertConformalConicProjection::getProj4ProjectionPart(std::ostream &) const` `[protected, virtual]`

add the pure projection parameters for proj4 to the stream, i.e. no earth definitions, and no +no_defs

Implements [MetNoFimex::ProjectionImpl](#).

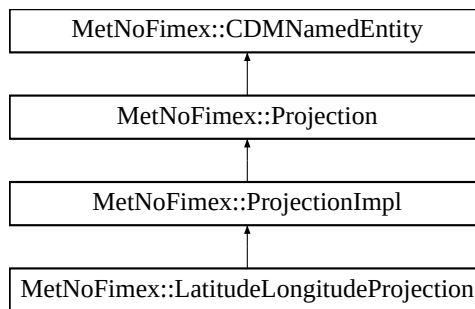
The documentation for this class was generated from the following file:

- `include/fimex/coordSys/LambertConformalConicProjection.h`

14.54 MetNoFimex::LatitudeLongitudeProjection Class Reference

```
#include <LatitudeLongitudeProjection.h>
```

Inheritance diagram for MetNoFimex::LatitudeLongitudeProjection::



Public Member Functions

- [LatitudeLongitudeProjection](#) ()
- virtual [~LatitudeLongitudeProjection](#) ()

Static Public Member Functions

- static bool [acceptsProj4](#) (const **std::string** &proj4Str)
- static **std::vector**< [CDMAAttribute](#) > [parametersFromProj4](#) (const **std::string** &proj4)

Protected Member Functions

- virtual **std::ostream** & [getProj4ProjectionPart](#) (**std::ostream** &oproj) const

14.54.1 Constructor & Destructor Documentation

14.54.1.1 `MetNoFimex::LatitudeLongitudeProjection::LatitudeLongitudeProjection ()`

14.54.1.2 `virtual MetNoFimex::LatitudeLongitudeProjection::~~LatitudeLongitudeProjection ()`
[inline, virtual]

14.54.2 Member Function Documentation

14.54.2.1 `static bool MetNoFimex::LatitudeLongitudeProjection::acceptsProj4 (const std::string & proj4Str)` [static]

14.54.2.2 `static std::vector<CDMAAttribute> MetNoFimex::LatitudeLongitudeProjection::parametersFromProj4 (const std::string & proj4)` [static]

14.54.2.3 `virtual std::ostream& MetNoFimex::LatitudeLongitudeProjection::getProj4ProjectionPart (std::ostream &) const` [inline, protected, virtual]

add the pure projection parameters for proj4 to the stream, i.e. no earth definitions, and no +no_defs

Implements [MetNoFimex::ProjectionImpl](#).

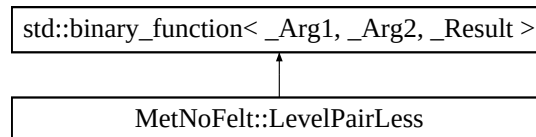
The documentation for this class was generated from the following file:

- `include/fimex/coordSys/LatitudeLongitudeProjection.h`

14.55 MetNoFelt::LevelPairLess Struct Reference

```
#include <Felt_Types.h>
```

Inheritance diagram for MetNoFelt::LevelPairLess::



Public Member Functions

- `bool operator() (const LevelPair &p1, const LevelPair &p2) const`

14.55.1 Detailed Description

comparison operator for `pair<short, short>` used for LevelPairs

14.55.2 Member Function Documentation

14.55.2.1 `bool MetNoFelt::LevelPairLess::operator() (const LevelPair &p1, const LevelPair &p2) const` `[inline]`

References `std::pair<_T1, _T2 >::first`, and `std::pair<_T1, _T2 >::second`.

The documentation for this struct was generated from the following file:

- `include/fimex/Felt_Types.h`

14.56 MetNoFimex::Logger Class Reference

```
#include <Logger.h>
```

Public Types

- enum [LogLevel](#) {
[OFF](#) = 1000, [FATAL](#) = 900, [ERROR](#) = 800, [WARN](#) = 700,
[INFO](#) = 600, [DEBUG](#) = 500 }

Public Member Functions

- [Logger](#) (const **std::string** &className)
- virtual [~Logger](#) ()
- virtual bool [isEnabledFor](#) ([LogLevel](#) level)
- virtual void [forcedLog](#) ([LogLevel](#) level, const **std::string** &message, const char *filename, unsigned int lineNumber)

14.56.1 Detailed Description

Interface and default (dummy) implementation for a logger. Don't use this class directly, but retrieve a pointer to it via the [getLogger](#) function and log with the [LOG4FIMEX](#) macro.

14.56.2 Member Enumeration Documentation

14.56.2.1 enum MetNoFimex::Logger::LogLevel

different log levels

Enumerator:

OFF
FATAL
ERROR
WARN
INFO
DEBUG

14.56.3 Constructor & Destructor Documentation

14.56.3.1 MetNoFimex::Logger::Logger (const **std::string** & *className*)

14.56.3.2 virtual MetNoFimex::Logger::~~Logger () [virtual]

14.56.4 Member Function Documentation

14.56.4.1 virtual bool MetNoFimex::Logger::isEnabledFor ([LogLevel](#) *level*) [virtual]

check if the loglevel of this logger is active

14.56.4.2 virtual void MetNoFimex::Logger::forcedLog (LogLevel *level*, const std::string & *message*, const char * *filename*, unsigned int *lineNumber*) [virtual]

log (without checking) for this loglevel

Parameters:

level log-level to log

message log-message

filename best retrieved with __FILE__

lineNumber best retrieved with __LINE__

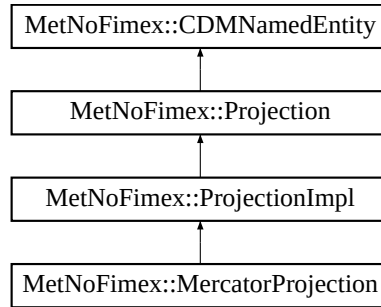
The documentation for this class was generated from the following file:

- include/fimex/[Logger.h](#)

14.57 MetNoFimex::MercatorProjection Class Reference

```
#include <MercatorProjection.h>
```

Inheritance diagram for MetNoFimex::MercatorProjection::



Public Member Functions

- [MercatorProjection](#) ()
- virtual [~MercatorProjection](#) ()

Static Public Member Functions

- static bool [acceptsProj4](#) (const **std::string** &proj4Str)
- static **std::vector**< [CDMAttribute](#) > [parametersFromProj4](#) (const **std::string** &proj4)

Protected Member Functions

- virtual **std::ostream** & [getProj4ProjectionPart](#) (**std::ostream** &oproj) const

14.57.1 Constructor & Destructor Documentation

14.57.1.1 MetNoFimex::MercatorProjection::MercatorProjection ()

14.57.1.2 virtual MetNoFimex::MercatorProjection::~~MercatorProjection () [virtual]

14.57.2 Member Function Documentation

14.57.2.1 static bool MetNoFimex::MercatorProjection::acceptsProj4 (const **std::string** &proj4Str) [static]

14.57.2.2 static **std::vector**<CDMAttribute> MetNoFimex::MercatorProjection::parametersFromProj4 (const **std::string** &proj4) [static]

14.57.2.3 virtual **std::ostream**& MetNoFimex::MercatorProjection::getProj4ProjectionPart (**std::ostream** &) const [protected, virtual]

add the pure projection parameters for proj4 to the stream, i.e. no earth definitions, and no +no_defs

Implements [MetNoFimex::ProjectionImpl](#).

The documentation for this class was generated from the following file:

- [include/fimex/coordSys/MercatorProjection.h](#)

14.58 mifi_cdm_reader Class Reference

```
#include <mifi_cdm_reader.h>
```

Public Member Functions

- [mifi_cdm_reader](#) (boost::shared_ptr< [MetNoFimex::CDMReader](#) > reader)
- boost::shared_ptr< [MetNoFimex::CDMReader](#) > [get](#) ()

14.58.1 Detailed Description

wrapper class for boost::shared_ptr<CDMReader>, mainly for usage by C/C++ wrapper

14.58.2 Constructor & Destructor Documentation

14.58.2.1 [mifi_cdm_reader::mifi_cdm_reader](#) (boost::shared_ptr< [MetNoFimex::CDMReader](#) > *reader*) `[inline]`

14.58.3 Member Function Documentation

14.58.3.1 [boost::shared_ptr<MetNoFimex::CDMReader> mifi_cdm_reader::get](#) () `[inline]`

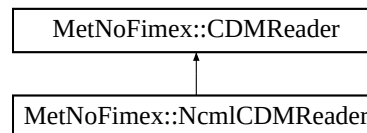
The documentation for this class was generated from the following file:

- [include/fimex/mifi_cdm_reader.h](#)

14.59 MetNoFimex::NcmlCDMReader Class Reference

```
#include <NcmlCDMReader.h>
```

Inheritance diagram for MetNoFimex::NcmlCDMReader::



Public Member Functions

- [NcmlCDMReader](#) (**std::string** configFile) throw (CDMException)
- [NcmlCDMReader](#) (const boost::shared_ptr< [CDMReader](#) > dataReader, **std::string** configFile) throw (CDMException)
- virtual [~NcmlCDMReader](#) ()
- virtual boost::shared_ptr< [Data](#) > [getDataSlice](#) (const **std::string** &varName, size_t unLimDimPos=0) throw (CDMException)

14.59.1 Detailed Description

The [NcmlCDMReader](#) can be used as both standard reader of a data and as a manipulator for an existing [CDM](#) provided by a [CDMReader](#).

In the case of a real reader, the ncml-configuration file needs to have the 'location' field set, which must point to a netcdf-file readable by NetCDF_CF10_CDMReader

The configuration file must be a standard ncml-file (versionn 2.2) as defined by <http://www.unidata.ucar.edu/software/netcdf/ncml/>.

Warning:

The current version does not support aggregation.

14.59.2 Constructor & Destructor Documentation

14.59.2.1 MetNoFimex::NcmlCDMReader::NcmlCDMReader (**std::string** *configFile*) throw (CDMException)

Parameters:

configFile ncml-file with location set

Exceptions:

[CDMException](#)

14.59.2.2 MetNoFimex::NcmlCDMReader::NcmlCDMReader (const boost::shared_ptr<CDMReader > *dataReader*, std::string *configFile*) throw (CDMException)

Parameters:

cdmReader a file reader opened elsewhere
configFile ncml-file with location set

Exceptions:

[CDMException](#)

14.59.2.3 virtual MetNoFimex::NcmlCDMReader::~~NcmlCDMReader () [virtual]

14.59.3 Member Function Documentation

14.59.3.1 virtual boost::shared_ptr<Data> MetNoFimex::NcmlCDMReader::getDataSlice (const std::string & *varName*, size_t *unLimDimPos* = 0) throw (CDMException) [virtual]

reading the data from the required source

Implements [MetNoFimex::CDMReader](#).

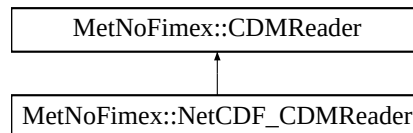
The documentation for this class was generated from the following file:

- include/fimex/[NcmlCDMReader.h](#)

14.60 MetNoFimex::NetCDF_CDMReader Class Reference

```
#include <fimex/NetCDF_CDMReader.h>
```

Inheritance diagram for MetNoFimex::NetCDF_CDMReader::



Public Member Functions

- [NetCDF_CDMReader](#) (const **std::string** &fileName)
- virtual [~NetCDF_CDMReader](#) ()
- virtual boost::shared_ptr< [Data](#) > [getDataSlice](#) (const **std::string** &varName, size_t unLimDimPos) throw (CDMException)
data-reading function to be called from the [CDMWriter](#)
- virtual boost::shared_ptr< [Data](#) > [getDataSlice](#) (const **std::string** &varName, const [SliceBuilder](#) &sb) throw (CDMException)
data-reading function to be called from the [CDMWriter](#)

14.60.1 Constructor & Destructor Documentation

14.60.1.1 MetNoFimex::NetCDF_CDMReader::NetCDF_CDMReader (const **std::string** & *fileName*)

14.60.1.2 virtual MetNoFimex::NetCDF_CDMReader::~~NetCDF_CDMReader () [virtual]

14.60.2 Member Function Documentation

14.60.2.1 virtual boost::shared_ptr<[Data](#)> MetNoFimex::NetCDF_CDMReader::getDataSlice (const **std::string** & *varName*, size_t *unLimDimPos*) throw (CDMException) [virtual]

data-reading function to be called from the [CDMWriter](#)

This methods needs to be implemented by the [CDMReader](#). It should provide the data for each variable, either by reading from disk, converting from another [CDMReader](#) or reading from an in-memory data-section.

This function should retrieve the whole data for a dataset without unlimited dimension if the unLimDimPos == 0.

Parameters:

varName name of the variable to read

unLimDimPos (optional) if the variable contains a unlimited dimension (max one allowed) an slice of this position is returned

Implements [MetNoFimex::CDMReader](#).

14.60.2.2 `virtual boost::shared_ptr<Data> MetNoFimex::NetCDF_CDMReader::getDataSlice
(const std::string & varName, const SliceBuilder & sb) throw (CDMException)
[virtual]`

data-reading function to be called from the [CDMWriter](#)

Parameters:

varName name of the variable to read

sb a [SliceBuilder](#) generated from this CDMReaders [CDM](#) This method has a default implementation depending on `getDataSlice(varName, unLimDimPos)`, but should be implemented for performance reasons.

Reimplemented from [MetNoFimex::CDMReader](#).

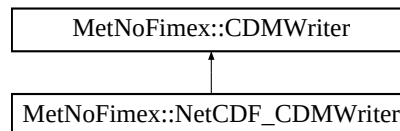
The documentation for this class was generated from the following file:

- `include/fimex/NetCDF_CDMReader.h`

14.61 MetNoFimex::NetCDF_CDMWriter Class Reference

```
#include <NetCDF_CDMWriter.h>
```

Inheritance diagram for MetNoFimex::NetCDF_CDMWriter::



Public Member Functions

- [NetCDF_CDMWriter](#) (const boost::shared_ptr< [CDMReader](#) > *cdmReader*, const std::string &*outputFile*, std::string *configFile*="", int *version*=3)
- virtual [~NetCDF_CDMWriter](#) ()
- const std::string & [getVariableName](#) (const std::string &*varName*) const
- const std::string & [getDimensionName](#) (const std::string &*dimName*) const
- const [CDMAttribute](#) & [getAttribute](#) (const std::string &*varName*, const std::string &*attName*) const throw (CDMException)

14.61.1 Constructor & Destructor Documentation

14.61.1.1 MetNoFimex::NetCDF_CDMWriter::NetCDF_CDMWriter (const boost::shared_ptr< CDMReader > *cdmReader*, const std::string & *outputFile*, std::string *configFile* = "", int *version* = 3)

Parameters:

cdmReader dataSource

outputFile file-name to write to

configFile xml-configuration

netcdf version, can be 3 or 4; 4 requires compilation against netcdf-4.0 or higher

14.61.1.2 virtual MetNoFimex::NetCDF_CDMWriter::~~NetCDF_CDMWriter () [virtual]

14.61.2 Member Function Documentation

14.61.2.1 const std::string& MetNoFimex::NetCDF_CDMWriter::getVariableName (const std::string & *varName*) const

Warning:

only public for testing

Returns:

the new name of a variable, eventually changed by the writers config

14.61.2.2 `const std::string& MetNoFimex::NetCDF_CDMWriter::getDimensionName (const std::string & dimName) const`

Warning:

only public for testing

Returns:

the new name of a dimension, eventually changed by the writers config

14.61.2.3 `const CDMAttribute& MetNoFimex::NetCDF_CDMWriter::getAttribute (const std::string & varName, const std::string & attName) const throw (CDMException)`

Warning:

only public for testing

Parameters:

varName original variable name (before config: newname)

attName original attribute name (before config: newname)

Returns:

an attribute contained in the writers attribute, possibly added by config

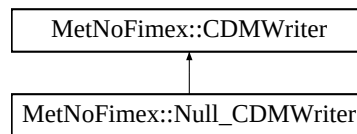
The documentation for this class was generated from the following file:

- include/fimex/[NetCDF_CDMWriter.h](#)

14.62 MetNoFimex::Null_CDMWriter Class Reference

```
#include <Null_CDMWriter.h>
```

Inheritance diagram for MetNoFimex::Null_CDMWriter::



Public Member Functions

- [Null_CDMWriter](#) (const boost::shared_ptr< [CDMReader](#) > [cdmReader](#), const std::string &[outputFile](#))
- virtual [~Null_CDMWriter](#) ()

14.62.1 Detailed Description

[CDMWriter](#) does all operations as the [NetCDF_CDMWriter](#), except writing to the file. This class is useful for performance tests.

14.62.2 Constructor & Destructor Documentation

14.62.2.1 MetNoFimex::Null_CDMWriter::Null_CDMWriter (const boost::shared_ptr< [CDMReader](#) > [cdmReader](#), const std::string & [outputFile](#))

14.62.2.2 virtual MetNoFimex::Null_CDMWriter::~~Null_CDMWriter () [virtual]

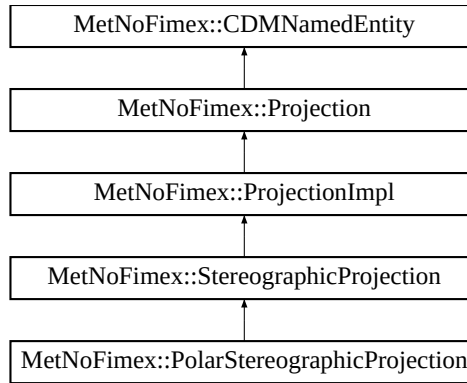
The documentation for this class was generated from the following file:

- include/fimex/[Null_CDMWriter.h](#)

14.63 MetNoFimex::PolarStereographicProjection Class Reference

```
#include <PolarStereographicProjection.h>
```

Inheritance diagram for MetNoFimex::PolarStereographicProjection::



Public Member Functions

- [PolarStereographicProjection \(\)](#)
- virtual [~PolarStereographicProjection \(\)](#)

Static Public Member Functions

- static bool [acceptsProj4](#) (const `std::string` &proj4Str)
- static `std::vector`< [CDMAttribute](#) > [parametersFromProj4](#) (const `std::string` &proj4)

14.63.1 Constructor & Destructor Documentation

14.63.1.1 `MetNoFimex::PolarStereographicProjection::PolarStereographicProjection ()`
[inline]

14.63.1.2 `virtual MetNoFimex::PolarStereographicProjection::~~PolarStereographicProjection ()`
[inline, virtual]

14.63.2 Member Function Documentation

14.63.2.1 `static bool MetNoFimex::PolarStereographicProjection::acceptsProj4 (const std::string &proj4Str)` [static]

Reimplemented from [MetNoFimex::StereographicProjection](#).

14.63.2.2 `static std::vector<CDMAttribute> MetNoFimex::PolarStereographicProjection::parametersFromProj4 (const std::string &proj4)` [static]

Reimplemented from [MetNoFimex::StereographicProjection](#).

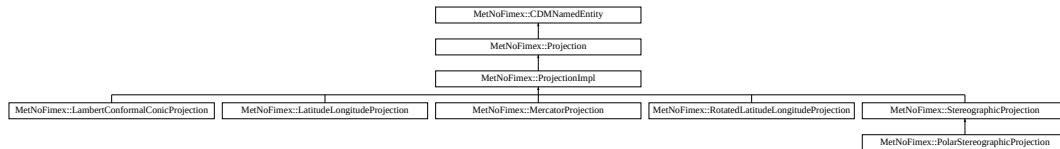
The documentation for this class was generated from the following file:

- `include/fimex/coordSys/PolarStereographicProjection.h`

14.64 MetNoFimex::Projection Class Reference

```
#include <Projection.h>
```

Inheritance diagram for MetNoFimex::Projection::



Public Member Functions

- virtual [~Projection](#) ()
- virtual **std::vector**< [CDMAttribute](#) > [getParameters](#) () const =0
- virtual void [addParameter](#) ([CDMAttribute](#) attribute)=0
- virtual void [addParameters](#) (**std::vector**< [CDMAttribute](#) > attributes)=0
- virtual void [removeParameter](#) (**std::string** paramName)=0
- virtual const **std::string** & [getName](#) () const =0
- virtual const bool [isDegree](#) () const =0
- virtual **std::string** [getProj4String](#) () const =0
- virtual **std::string** [toString](#) () const =0
- virtual bool [operator==](#) (const [Projection](#) &b) const

Static Public Member Functions

- static boost::shared_ptr< [Projection](#) > [create](#) (**std::vector**< [CDMAttribute](#) >)
- static boost::shared_ptr< [Projection](#) > [createByProj4](#) (const **std::string** &projStr)

Protected Member Functions

- [Projection](#) ()

14.64.1 Detailed Description

A projection describes a projection of the earth surface from one system to another. If not mentioned otherwise, all parameters should follow CF-1.x

14.64.2 Constructor & Destructor Documentation

14.64.2.1 `virtual MetNoFimex::Projection::~~Projection () [inline, virtual]`

14.64.2.2 `MetNoFimex::Projection::Projection () [inline, protected]`

14.64.3 Member Function Documentation

14.64.3.1 `virtual std::vector<CDMAAttribute> MetNoFimex::Projection::getParameters () const [pure virtual]`

Implemented in [MetNoFimex::ProjectionImpl](#).

14.64.3.2 `virtual void MetNoFimex::Projection::addParameter (CDMAAttribute attribute) [pure virtual]`

Implemented in [MetNoFimex::ProjectionImpl](#).

14.64.3.3 `virtual void MetNoFimex::Projection::addParameters (std::vector< CDMAAttribute > attributes) [pure virtual]`

Implemented in [MetNoFimex::ProjectionImpl](#).

14.64.3.4 `virtual void MetNoFimex::Projection::removeParameter (std::string paramName) [pure virtual]`

Implemented in [MetNoFimex::ProjectionImpl](#).

14.64.3.5 `virtual const std::string& MetNoFimex::Projection::getName () const [pure virtual]`

get the projection name

Implements [MetNoFimex::CDMNamedEntity](#).

Implemented in [MetNoFimex::ProjectionImpl](#).

14.64.3.6 `virtual const bool MetNoFimex::Projection::isDegree () const [pure virtual]`

check if the coordinates belonging to this projection are in degree (otherwise metrical)

Implemented in [MetNoFimex::ProjectionImpl](#).

14.64.3.7 `virtual std::string MetNoFimex::Projection::getProj4String () const [pure virtual]`

get a proj4 string

Implemented in [MetNoFimex::ProjectionImpl](#).

14.64.3.8 `virtual std::string MetNoFimex::Projection::toString () const` [pure virtual]

get a string representation

Note:

this should be implemented as unique as possible, i.e.

Implemented in [MetNoFimex::ProjectionImpl](#).

14.64.3.9 `virtual bool MetNoFimex::Projection::operator== (const Projection & b) const`
[virtual]

Comparison of two projections, implemented using the [toString\(\)](#) function. This function does not guarantee that two projections are physically equal.

14.64.3.10 `static boost::shared_ptr<Projection> MetNoFimex::Projection::create (std::vector<CDMAAttribute >) [static]`

create a projection from some CDMAAttributes

14.64.3.11 `static boost::shared_ptr<Projection> MetNoFimex::Projection::createByProj4 (const std::string & projStr) [static]`

create a projection from a proj4 string

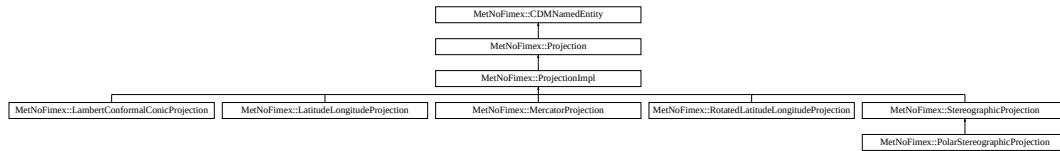
The documentation for this class was generated from the following file:

- [include/fimex/coordSys/Projection.h](#)

14.65 MetNoFimex::ProjectionImpl Class Reference

```
#include <ProjectionImpl.h>
```

Inheritance diagram for MetNoFimex::ProjectionImpl::



Public Member Functions

- virtual `~ProjectionImpl()`
- virtual `std::vector< CDMAttribute > getParameters()` const
- virtual void `addParameter(CDMAttribute attribute)`
- virtual void `addParameters(std::vector< CDMAttribute > attributes)`
- virtual void `removeParameter(std::string paramName)`
- virtual const `std::string & getName()` const
- virtual const bool `isDegree()` const
- virtual `std::string getProj4String()` const
- virtual `std::string toString()` const

Protected Member Functions

- `ProjectionImpl(std::string name, bool isDegree)`
- virtual `std::ostream & getProj4ProjectionPart(std::ostream &) const =0`
- bool `addParameterToStream(std::ostream &outStream, const std::string &name, std::string replaceName="") const`

Static Protected Member Functions

- static bool `proj4ProjectionMatchesName(const std::string &proj4String, const std::string &name)`
- static void `proj4GetEarthAttributes(const std::string &proj4String, std::vector< CDMAttribute > &attrList)`

Protected Attributes

- `std::vector< CDMAttribute > params_`

14.65.1 Detailed Description

`ProjectionImpl` is a next to complete implementation of `Projection`, storing all parameters as a `vector<CDMAttribute>`. Implementations only need to implement a constructor calling the `ProjectionImpl("proj-name")` and the method `getProj4ProjectionPart()`

Note:

the implemented projection needs to get made visible in `Projection::create()` and `Projection::createByProj4`

14.65.2 Constructor & Destructor Documentation

14.65.2.1 `virtual MetNoFimex::ProjectionImpl::~~ProjectionImpl ()` [virtual]

14.65.2.2 `MetNoFimex::ProjectionImpl::ProjectionImpl (std::string name, bool isDegree)`
[explicit, protected]

14.65.3 Member Function Documentation

14.65.3.1 `virtual std::vector<CDMAAttribute> MetNoFimex::ProjectionImpl::getParameters ()`
`const` [virtual]

Implements [MetNoFimex::Projection](#).

14.65.3.2 `virtual void MetNoFimex::ProjectionImpl::addParameter (CDMAAttribute attribute)`
[virtual]

Implements [MetNoFimex::Projection](#).

14.65.3.3 `virtual void MetNoFimex::ProjectionImpl::addParameters (std::vector< CDMAAttribute > attributes)` [virtual]

Implements [MetNoFimex::Projection](#).

14.65.3.4 `virtual void MetNoFimex::ProjectionImpl::removeParameter (std::string paramName)`
[virtual]

Implements [MetNoFimex::Projection](#).

14.65.3.5 `virtual const std::string& MetNoFimex::ProjectionImpl::getName () const`
[virtual]

get the projection name

Implements [MetNoFimex::Projection](#).

14.65.3.6 `virtual const bool MetNoFimex::ProjectionImpl::isDegree () const` [virtual]

check if the coordinates belonging to this projection are in degree (otherwise metrical)

Implements [MetNoFimex::Projection](#).

14.65.3.7 `virtual std::string MetNoFimex::ProjectionImpl::getProj4String () const` [virtual]

get the proj4 string defined by the parameters. If a parameter named 'proj4' exists, that one will be used and all other parameters will be ignored.

Note:

Implementors should not overwrite this method, but the protected [getProj4ProjectionPart\(\)](#) method

Implements [MetNoFimex::Projection](#).

14.65.3.8 virtual std::string MetNoFimex::ProjectionImpl::toString () const [virtual]

get a string representation

Implements [MetNoFimex::Projection](#).

14.65.3.9 static bool MetNoFimex::ProjectionImpl::proj4ProjectionMatchesName (const std::string & *proj4String*, const std::string & *name*) [static, protected]

match the +proj= part of a proj4 string

14.65.3.10 static void MetNoFimex::ProjectionImpl::proj4GetEarthAttributes (const std::string & *proj4String*, std::vector< CDMAAttribute > & *attrList*) [static, protected]

add the attributes describing the earth from a proj4-string to the outAttrs

Parameters:

proj4String string as used for proj4

output list of CDMAAttributes

14.65.3.11 virtual std::ostream& MetNoFimex::ProjectionImpl::getProj4ProjectionPart (std::ostream &) const [protected, pure virtual]

add the pure projection parameters for proj4 to the stream, i.e. no earth definitions, and no +no_defs

Implemented in [MetNoFimex::LambertConformalConicProjection](#), [MetNoFimex::LatitudeLongitudeProjection](#), [MetNoFimex::MercatorProjection](#), [MetNoFimex::RotatedLatitudeLongitudeProjection](#), and [MetNoFimex::StereographicProjection](#).

14.65.3.12 bool MetNoFimex::ProjectionImpl::addParameterToStream (std::ostream & *outStream*, const std::string & *name*, std::string *replaceName* = "") const [protected]

Add the numeric value of a parameter named *name* as *replaceName* to *oproj*, e.g. *name* = false_easting, *replaceName* = +x_0, sets "+x_0=..." Assume only one value at maximum

Parameters:

outStream

name the parameters name

replaceName the name to use in the stream to the parameter, defaults to original name ("")

Returns:

true if parameter found and set

14.65.4 Member Data Documentation

14.65.4.1 `std::vector<CDMAAttribute> MetNoFimex::ProjectionImpl::params_` [protected]

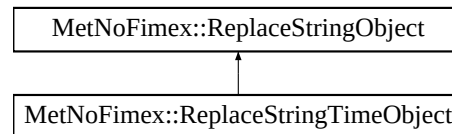
The documentation for this class was generated from the following file:

- `include/fimex/coordSys/ProjectionImpl.h`

14.66 MetNoFimex::ReplaceStringObject Class Reference

```
#include <ReplaceStringObject.h>
```

Inheritance diagram for MetNoFimex::ReplaceStringObject::



Public Member Functions

- virtual [~ReplaceStringObject](#) ()=0
- virtual `std::ostream & put (std::ostream &s) const` =0
- virtual void [setFormatString](#) (const `std::string` &format)=0
set the formatting string for this object
- virtual void [setFormatStringAndOptions](#) (const `std::string` &format, const `std::vector< std::string >` &options)=0
set the formatting string and additional options for this object

14.66.1 Detailed Description

Interface for objects which might be converted to different strings

14.66.2 Constructor & Destructor Documentation

14.66.2.1 virtual `MetNoFimex::ReplaceStringObject::~~ReplaceStringObject ()` [pure virtual]

14.66.3 Member Function Documentation

14.66.3.1 virtual `std::ostream& MetNoFimex::ReplaceStringObject::put (std::ostream & s) const` [pure virtual]

put the formatted string to the stream

implementors are asked to implement operator<<

Implemented in [MetNoFimex::ReplaceStringTimeObject](#).

14.66.3.2 virtual void `MetNoFimex::ReplaceStringObject::setFormatString (const std::string &format)` [pure virtual]

set the formatting string for this object

Implemented in [MetNoFimex::ReplaceStringTimeObject](#).

14.66.3.3 `virtual void MetNoFimex::ReplaceStringObject::setFormatStringAndOptions (const std::string & format, const std::vector< std::string > & options)` [pure virtual]

set the formatting string and additional options for this object

Implemented in [MetNoFimex::ReplaceStringTimeObject](#).

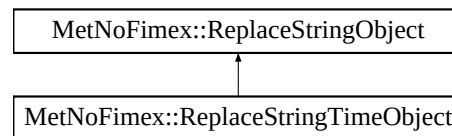
The documentation for this class was generated from the following file:

- [include/fimex/ReplaceStringObject.h](#)

14.67 MetNoFimex::ReplaceStringTimeObject Class Reference

```
#include <ReplaceStringTimeObject.h>
```

Inheritance diagram for MetNoFimex::ReplaceStringTimeObject::



Public Member Functions

- [ReplaceStringTimeObject](#) ()
- [ReplaceStringTimeObject](#) (std::time_t time, std::string format="%Y-%m-%d %H:%M:%S%F%Q")
- virtual [~ReplaceStringTimeObject](#) ()
- virtual std::ostream & [put](#) (std::ostream &s) const
- virtual void [setFormatString](#) (const std::string &format)
- virtual void [setFormatStringAndOptions](#) (const std::string &format, const std::vector< std::string > &options)

Friends

- std::ostream & [operator<<](#) (std::ostream &s, const [ReplaceStringTimeObject](#) &rsto)

14.67.1 Constructor & Destructor Documentation

14.67.1.1 MetNoFimex::ReplaceStringTimeObject::ReplaceStringTimeObject () [inline]

14.67.1.2 MetNoFimex::ReplaceStringTimeObject::ReplaceStringTimeObject (std::time_t time, std::string format = "%Y-%m-%d %H:%M:%S%F%Q") [inline]

initialize a [ReplaceStringTimeObject](#) with time and string set

14.67.1.3 virtual MetNoFimex::ReplaceStringTimeObject::~~ReplaceStringTimeObject () [inline, virtual]

14.67.2 Member Function Documentation

14.67.2.1 virtual std::ostream& MetNoFimex::ReplaceStringTimeObject::put (std::ostream &s) const [inline, virtual]

put the formatted string to the stream

implementors are asked to implement operator<<

Implements [MetNoFimex::ReplaceStringObject](#).

14.67.2.2 **virtual void MetNoFimex::ReplaceStringTimeObject::setFormatString (const std::string & *format*)** [inline, virtual]

set the formatting String for this object

Parameters:

format,: format string of strftime <http://www.cplusplus.com/reference/clibrary/ctime/strftime>.

Implements [MetNoFimex::ReplaceStringObject](#).

14.67.2.3 **virtual void MetNoFimex::ReplaceStringTimeObject::setFormatStringAndOptions (const std::string & *format*, const std::vector< std::string > & *options*)** [virtual]

set the formatting string and additional options for this object options are: 0: offset as in seconds, i.e. +5000, -6000

Implements [MetNoFimex::ReplaceStringObject](#).

14.67.3 Friends And Related Function Documentation

14.67.3.1 **std::ostream& operator<< (std::ostream & *s*, const ReplaceStringTimeObject & *rsto*)** [friend]

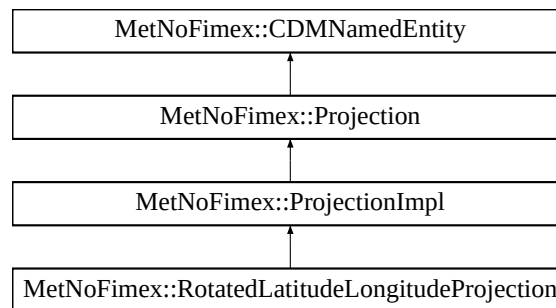
The documentation for this class was generated from the following file:

- include/fimex/[ReplaceStringTimeObject.h](#)

14.68 MetNoFimex::RotatedLatitudeLongitudeProjection Class Reference

```
#include <RotatedLatitudeLongitudeProjection.h>
```

Inheritance diagram for MetNoFimex::RotatedLatitudeLongitudeProjection::



Public Member Functions

- [RotatedLatitudeLongitudeProjection](#) ()
- virtual [~RotatedLatitudeLongitudeProjection](#) ()

Static Public Member Functions

- static bool [acceptsProj4](#) (const **std::string** &proj4Str)
- static **std::vector**< [CDMAttribute](#) > [parametersFromProj4](#) (const **std::string** &proj4)

Protected Member Functions

- virtual **std::ostream** & [getProj4ProjectionPart](#) (**std::ostream** &oproj) const

14.68.1 Constructor & Destructor Documentation

14.68.1.1 `MetNoFimex::RotatedLatitudeLongitudeProjection::RotatedLatitudeLongitudeProjection()`

14.68.1.2 `virtual MetNoFimex::RotatedLatitudeLongitudeProjection::~~RotatedLatitudeLongitudeProjection()` `[inline, virtual]`

14.68.2 Member Function Documentation

14.68.2.1 `static bool MetNoFimex::RotatedLatitudeLongitudeProjection::acceptsProj4 (const std::string & proj4Str)` `[static]`

14.68.2.2 `static std::vector<CDMAAttribute> MetNoFimex::RotatedLatitudeLongitudeProjection::parametersFromProj4 (const std::string & proj4)` `[static]`

14.68.2.3 `virtual std::ostream& MetNoFimex::RotatedLatitudeLongitudeProjection::getProj4ProjectionPart (std::ostream &) const` `[protected, virtual]`

add the pure projection parameters for proj4 to the stream, i.e. no earth definitions, and no +no_defs

Implements [MetNoFimex::ProjectionImpl](#).

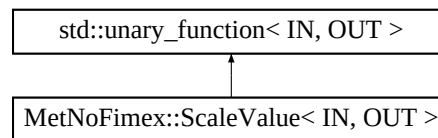
The documentation for this class was generated from the following file:

- `include/fimex/coordSys/RotatedLatitudeLongitudeProjection.h`

14.69 MetNoFimex::ScaleValue< IN, OUT > Class Template Reference

```
#include <Utils.h>
```

Inheritance diagram for MetNoFimex::ScaleValue< IN, OUT >::



Public Member Functions

- [ScaleValue](#) (double *oldFill*, double *oldScale*, double *oldOffset*, double *newFill*, double *newScale*, double *newOffset*)
- OUT [operator\(\)](#) (const IN &*in*) const

14.69.1 Detailed Description

```
template<typename IN, typename OUT> class MetNoFimex::ScaleValue< IN, OUT >
```

Scale a value using fill, offset and scale

14.69.2 Constructor & Destructor Documentation

14.69.2.1 `template<typename IN, typename OUT> MetNoFimex::ScaleValue< IN, OUT >::ScaleValue (double oldFill, double oldScale, double oldOffset, double newFill, double newScale, double newOffset) [inline]`

14.69.3 Member Function Documentation

14.69.3.1 `template<typename IN, typename OUT> OUT MetNoFimex::ScaleValue< IN, OUT >::operator() (const IN &in) const [inline]`

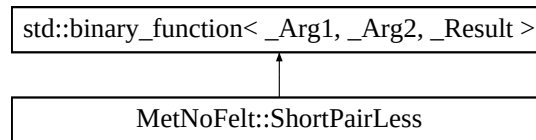
The documentation for this class was generated from the following file:

- include/fimex/[Utils.h](#)

14.70 MetNoFelt::ShortPairLess Struct Reference

```
#include <Felt_Array.h>
```

Inheritance diagram for MetNoFelt::ShortPairLess::



Public Member Functions

- `bool operator()` (const `pair< short, short >` &p1, const `pair< short, short >` &p2) const

14.70.1 Detailed Description

comparison operator for `pair<short, short>` used for levelPairs

14.70.2 Member Function Documentation

14.70.2.1 `bool MetNoFelt::ShortPairLess::operator()` (const `pair< short, short >` &p1, const `pair< short, short >` &p2) const `[inline]`

References `std::pair<_T1, _T2 >::first`, and `std::pair<_T1, _T2 >::second`.

The documentation for this struct was generated from the following file:

- `include/fimex/Felt_Array.h`

14.71 MetNoFimex::SliceBuilder Class Reference

```
#include <fimex/SliceBuilder.h>
```

Public Member Functions

- [SliceBuilder](#) (const [CDM](#) &cdm, const **std::string** &varName)
- [~SliceBuilder](#) ()
- void [setStartAndSize](#) (const **std::string** &dimName, size_t start, size_t size)
- void [setStartAndSize](#) (const boost::shared_ptr< const [CoordinateAxis](#) > &axis, size_t start, size_t size)
- const **std::vector**< size_t > & [getDimensionStartPositions](#) () const
- const **std::vector**< size_t > & [getDimensionSizes](#) () const
- **std::vector**< **std::string** > [getDimensionNames](#) () const
- const **std::vector**< size_t > & [getMaxDimensionSizes](#) () const

14.71.1 Constructor & Destructor Documentation

14.71.1.1 MetNoFimex::SliceBuilder::SliceBuilder (const CDM & cdm, const std::string & varName)

Create a new slice builder for variable varName. The default will give a slice of full size.

Parameters:

cdm
varName variable name

Exceptions:

[CDMException](#) if varName doesn't exists

14.71.1.2 MetNoFimex::SliceBuilder::~~SliceBuilder ()

14.71.2 Member Function Documentation

14.71.2.1 void MetNoFimex::SliceBuilder::setStartAndSize (const std::string & dimName, size_t start, size_t size)

set the start position and the size of the slice of that dimension

Parameters:

dimName name of the dimension to restrict
start starting point of slice (starts at 0)
size size of the slice

Exceptions:

[CDMException](#) if dimName not part of the dimensions of variable
out_of_range depending on startPos or size

14.71.2.2 void MetNoFimex::SliceBuilder::setStartAndSize (const boost::shared_ptr< const CoordinateAxis > & axis, size_t start, size_t size)

set the start position and the size of the slice of that dimension

Parameters:

axis name of the dimension to restrict, ignored if NULL
start starting point of slice (starts at 0)
size size of the slice

Exceptions:

CDMException if axis not part of the dimensions of the variable
out_of_range depending on startPos or size

14.71.2.3 const std::vector<size_t> & MetNoFimex::SliceBuilder::getDimensionStartPositions () const [inline]

Returns:

vector with start-positions of shape-size and order of the variable

14.71.2.4 const std::vector<size_t> & MetNoFimex::SliceBuilder::getDimensionSizes () const [inline]

Returns:

vector with sizes of the dimensions of the variable in the order and size of the variables dimensions

14.71.2.5 std::vector<std::string> MetNoFimex::SliceBuilder::getDimensionNames () const

Returns:

vector with names of dimensions in correct order

14.71.2.6 const std::vector<size_t> & MetNoFimex::SliceBuilder::getMaxDimensionSizes () const [inline]

Returns:

vector with maximum sizes of the dimensions of the variable in the order and size of the variables dimensions

The documentation for this class was generated from the following file:

- include/fimex/[SliceBuilder.h](#)

14.72 MetNoFimex::SpatialAxisSpec Class Reference

```
#include <SpatialAxisSpec.h>
```

Public Member Functions

- [SpatialAxisSpec](#) (const **std::string** &axisSpec) throw (CDMException)
- [SpatialAxisSpec](#) (const **std::string** &axisSpec, double start, double end) throw (CDMException)
- virtual [~SpatialAxisSpec](#) ()
- bool [requireStartEnd](#) ()
- void [setStartEnd](#) (double start, double end)
- const **std::vector**< double > & [getAxisSteps](#) ()

14.72.1 Detailed Description

This class can be used to describe a list of spatial units in an efficient textual way.

- UNIT: see [udunit](#), compatible with degree or m: default: m
- RELVALUE: float-number
- RELVALUES: comma-separated list of values with possible ... extension, ... meaning continuation of the difference of the previous two values 0 is the first time in the original time-axis, x is the last time-value in the original time-axis

A [SpatialAxisSpec](#) consists of at least of values:

- axisSpec := VALUES[;unit=UNIT] | RELVALUES;relativeStart=VALUE[;unit=UNIT]

relativeStart will reset the relative value 0 to the first value larger than x0 (original start time) with $x_0 = i * (v_1 - v_0) * \text{unit}$ with i being a integer.

```
axisSpec = -450000,-400000,...,50000
```

```
timespec = -50,0,...,x,x+50;relativeStart=17;unit=km
```

Warning:

The 'unit' parameter is currently not supported, please enter values as m or degree
the RELVALUES currently must be in m, degree not supported (yet?)

14.72.2 Constructor & Destructor Documentation

14.72.2.1 MetNoFimex::SpatialAxisSpec::SpatialAxisSpec (const **std::string** & *axisSpec*) throw (CDMException) [inline]

Define a spatialAxisSpec. Depending on the axisSpec (relativeStart?), start and end must be given later

Parameters:

axisSpec string representation as explained above

14.72.2.2 MetNoFimex::SpatialAxisSpec::SpatialAxisSpec (const std::string & *axisSpec*, double *start*, double *end*) throw (CDMException) [inline]

Define a spatialAxisSpec

Parameters:

axisSpec string representation as explained above
start place of data start, in degree or m
end place to end, in degree or m

14.72.2.3 virtual MetNoFimex::SpatialAxisSpec::~~SpatialAxisSpec () [inline, virtual]

14.72.3 Member Function Documentation

14.72.3.1 bool MetNoFimex::SpatialAxisSpec::requireStartEnd ()

Check if axisSpec still requires start and end place. This returns false if a) start and end have been given already b) the axisSpec is independant of start and end

14.72.3.2 void MetNoFimex::SpatialAxisSpec::setStartEnd (double *start*, double *end*) [inline]

14.72.3.3 const std::vector<double>& MetNoFimex::SpatialAxisSpec::getAxisSteps () [inline]

Returns:

steps on the axis in degree or m

The documentation for this class was generated from the following file:

- [include/fimex/SpatialAxisSpec.h](#)

14.73 MetNoFimex::staticCast< OUT > Struct Template Reference

```
#include <Utils.h>
```

Public Member Functions

- `template<typename IN>`
`OUT operator\(\) (const IN &in)`

14.73.1 Detailed Description

`template<typename OUT> struct MetNoFimex::staticCast< OUT >`

`static_cast` as a functor

14.73.2 Member Function Documentation

14.73.2.1 `template<typename OUT> template<typename IN> OUT MetNoFimex::staticCast< OUT >::operator() (const IN & in)` `[inline]`

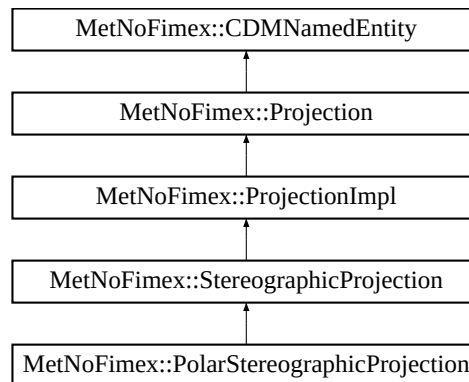
The documentation for this struct was generated from the following file:

- `include/fimex/Utils.h`

14.74 MetNoFimex::StereographicProjection Class Reference

```
#include <StereographicProjection.h>
```

Inheritance diagram for MetNoFimex::StereographicProjection::



Public Member Functions

- [StereographicProjection \(\)](#)
- virtual [~StereographicProjection \(\)](#)

Static Public Member Functions

- static bool [acceptsProj4](#) (const **std::string** &proj4Str)
- static **std::vector**< [CDMAAttribute](#) > [parametersFromProj4](#) (const **std::string** &proj4)

Protected Member Functions

- [StereographicProjection](#) (**std::string** name)
- virtual **std::ostream** & [getProj4ProjectionPart](#) (**std::ostream** &oproj) const

14.74.1 Constructor & Destructor Documentation

14.74.1.1 MetNoFimex::StereographicProjection::StereographicProjection ()

14.74.1.2 virtual MetNoFimex::StereographicProjection::~~StereographicProjection ()
[inline, virtual]

14.74.1.3 MetNoFimex::StereographicProjection::StereographicProjection (**std::string** name)
[inline, protected]

14.74.2 Member Function Documentation

14.74.2.1 static bool MetNoFimex::StereographicProjection::acceptsProj4 (const **std::string** &proj4Str) [static]

Reimplemented in [MetNoFimex::PolarStereographicProjection](#).

14.74.2.2 `static std::vector<CDMAAttribute> MetNoFimex::StereographicProjection::parametersFromProj4 (const std::string & proj4)` `[static]`

Reimplemented in [MetNoFimex::PolarStereographicProjection](#).

14.74.2.3 `virtual std::ostream& MetNoFimex::StereographicProjection::getProj4ProjectionPart (std::ostream &) const` `[protected, virtual]`

add the pure projection parameters for proj4 to the stream, i.e. no earth definitions, and no +no_defs

Implements [MetNoFimex::ProjectionImpl](#).

The documentation for this class was generated from the following file:

- `include/fimex/coordSys/StereographicProjection.h`

14.75 MetNoFimex::TimeLevelDataSliceFetcher Class Reference

read a slice of a given time/level combination from a cdmReader

```
#include <TimeLevelDataSliceFetcher.h>
```

Public Member Functions

- [TimeLevelDataSliceFetcher](#) (boost::shared_ptr< [CDMReader](#) > cdmReader, const std::string &varName)
- virtual [~TimeLevelDataSliceFetcher](#) ()
- boost::shared_ptr< [Data](#) > [getTimeLevelSlice](#) (size_t time, size_t level) throw (CDMException)

14.75.1 Detailed Description

read a slice of a given time/level combination from a cdmReader

14.75.2 Constructor & Destructor Documentation

14.75.2.1 MetNoFimex::TimeLevelDataSliceFetcher::TimeLevelDataSliceFetcher (boost::shared_ptr< CDMReader > cdmReader, const std::string & varName)

initialize the Fetcher

Parameters:

cdmReader the reader to fetch the original data from
varName the variable to read the data from

14.75.2.2 virtual MetNoFimex::TimeLevelDataSliceFetcher::~~TimeLevelDataSliceFetcher () [virtual]

14.75.3 Member Function Documentation

14.75.3.1 boost::shared_ptr<Data> MetNoFimex::TimeLevelDataSliceFetcher::getTimeLevelSlice (size_t time, size_t level) throw (CDMException)

get the slice of time at position time and level at position level join unlimited dimensions if needed, slice data if needed

Parameters:

time the position of the time according to the variables level-dimension
level the position of the level according to the level-dimension

The documentation for this class was generated from the following file:

- include/fimex/[TimeLevelDataSliceFetcher.h](#)

14.76 MetNoFimex::TimeSpec Class Reference

```
#include <TimeSpec.h>
```

Public Member Functions

- [TimeSpec](#) (const **std::string** &timeSpec, const [FimexTime](#) &startTime, const [FimexTime](#) &endTime) throw (CDMException)
- virtual [~TimeSpec](#) ()
- const **std::vector**< [FimexTime](#) > & [getTimeSteps](#) () const
- const **std::string** & [getUnitString](#) () const

14.76.1 Detailed Description

This class can be used to describe a list of times in an efficient textual way.

Unless otherwise mentioned, i.e. with *bounds* a value v(time) describes the time at exactly that instance. All times are UTC.

- **TIMESTAMP** format: YYYY-MM-DD HH:MM:SS
- **TIMESTAMPS**: comma-separated list of values with possible ... extension, ... meaning continuation of the difference of the previous two values
- **UNIT**: see [udunit](#), default: second
- **VALUE**: float-number
- **VALUES**: comma-separated list of values with possible ... extension, ... meaning continuation of the difference of the previous two values 0 is the first time in the original time-axis, x is the last time-value in the original time-axis

A [TimeSpec](#) consists of at least of timestamps or values:

- timespec := (TIMESTAMPS | VALUES[;relativeUnit=UNIT])[;unit=UNIT]

relativeUnit will reset the relative value 0 to the first value larger than t0 (original start time) with $t0 = i * (v1 - v0) * \text{unit}$ with i being a integer.

```
timespec = 2000-01-01 00:00:00,2000-01-01 00:04:00,...,2010-01-01 00:00:00
```

All times outside the original time-axis will be discarded.

```
timespec = -3,0,3,...,x,x+3;relativeUnit=hours since 2000-01-01 00:00:00;unit=hours since 2000-01-01 00:00:00
```

14.76.2 Constructor & Destructor Documentation

14.76.2.1 MetNoFimex::TimeSpec::TimeSpec (const **std::string** & *timeSpec*, const [FimexTime](#) & *startTime*, const [FimexTime](#) & *endTime*) throw (CDMException)

Define a timeSpec

Parameters:

timeSpec string representation as explained above
startTime time to start in case of a relativeStart timeSpec
endTime time to end in case of a relativeStart timeSpec

14.76.2.2 `virtual MetNoFimex::TimeSpec::~~TimeSpec ()` `[inline, virtual]`

14.76.3 Member Function Documentation

14.76.3.1 `const std::vector<FimexTime>& MetNoFimex::TimeSpec::getTimeSteps () const`
`[inline]`

14.76.3.2 `const std::string& MetNoFimex::TimeSpec::getUnitString () const` `[inline]`

The documentation for this class was generated from the following file:

- `include/fimex/TimeSpec.h`

14.77 MetNoFimex::TimeUnit Class Reference

```
#include <TimeUnit.h>
```

Public Member Functions

- [TimeUnit](#) () throw (CDMException)
initialize a timeUnit with a unit string
- [TimeUnit](#) (const **std::string** &timeUnitString) throw (CDMException)
- virtual [~TimeUnit](#) ()
- double [unitTime2epochSeconds](#) (double unitTime) const
calculate the epochSeconds for a time in the current unit
- double [epochSeconds2unitTime](#) (double epochSeconds) const
calculate the time in the current unit from the epoch
- [FimexTime](#) [unitTime2fimexTime](#) (double unitTime) const throw (CDMException)
calculate the time in a calendar form
- double [fimexTime2unitTime](#) (const [FimexTime](#) &fiTime) const throw (CDMException)
calculate the time in the current unit from the calendar form
- double [fimexTime2unitTimeX](#) ([FimexTime](#) fiTime) const throw (CDMException)
same as [fimexTime2unitTime](#) but copying fiTime instead of referencing, needed for i.e. `bind1st(mem_fun())`

14.77.1 Detailed Description

[TimeUnit](#) calculates times from a time given in a unit as of CF-1.0 (e.g. 'days since 2000-01-01 00:00:00') to a unix time (i.e. 'seconds since 1970-01-01 00:00:00') or a time struct [MetNoFimex::FimexTime](#)

All times are assumed to be UTC, and we use the Gregorian Calendar (not 100% true for times before 1600AD, depending on implementation)

14.77.2 Constructor & Destructor Documentation

14.77.2.1 MetNoFimex::TimeUnit::TimeUnit () throw (CDMException)

initialize a timeUnit with a unit string

14.77.2.2 `MetNoFimex::TimeUnit::TimeUnit (const std::string & timeUnitString) throw (CDMException)`

14.77.2.3 `virtual MetNoFimex::TimeUnit::~~TimeUnit () [virtual]`

14.77.3 Member Function Documentation

14.77.3.1 `double MetNoFimex::TimeUnit::unitTime2epochSeconds (double unitTime) const`

calculate the epochSeconds for a time in the current unit

14.77.3.2 `double MetNoFimex::TimeUnit::epochSeconds2unitTime (double epochSeconds) const`

calculate the time in the current unit from the epoch

14.77.3.3 `FimexTime MetNoFimex::TimeUnit::unitTime2fimexTime (double unitTime) const throw (CDMException)`

calculate the time in a calendar form

14.77.3.4 `double MetNoFimex::TimeUnit::fimexTime2unitTime (const FimexTime & fiTime) const throw (CDMException)`

calculate the time in the current unit from the calendar form

Referenced by `fimexTime2unitTimeX()`.

14.77.3.5 `double MetNoFimex::TimeUnit::fimexTime2unitTimeX (FimexTime fiTime) const throw (CDMException) [inline]`

same as `fimexTime2unitTime` but copying `fiTime` instead of referencing, needed for i.e. `bind1st(mem_fun())`

References `fimexTime2unitTime()`.

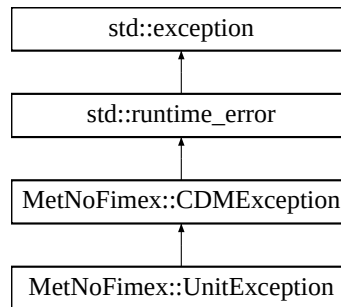
The documentation for this class was generated from the following file:

- `include/fimex/TimeUnit.h`

14.78 MetNoFimex::UnitException Class Reference

```
#include <Units.h>
```

Inheritance diagram for MetNoFimex::UnitException::



Public Member Functions

- [UnitException](#) ()
- [UnitException](#) (std::string message)

14.78.1 Constructor & Destructor Documentation

14.78.1.1 MetNoFimex::UnitException::UnitException () [inline]

14.78.1.2 MetNoFimex::UnitException::UnitException (std::string *message*) [inline]

The documentation for this class was generated from the following file:

- include/fimex/[Units.h](#)

14.79 MetNoFimex::Units Class Reference

```
#include <Units.h>
```

Public Member Functions

- [Units](#) ()
- [Units](#) (const [Units](#) &rhs)
- [Units](#) & [operator=](#) (const [Units](#) &rhs)
- virtual [~Units](#) ()
- void [convert](#) (const **std::string** &from, const **std::string** &to, double &slope, double &offset) throw (UnitException)
- bool [areConvertible](#) (const **std::string** &unit1, const **std::string** &unit2) const throw (UnitException)
test if two units are convertible to each others
- bool [isTime](#) (const **std::string** &timeUnit) const throw (UnitException)

14.79.1 Constructor & Destructor Documentation

14.79.1.1 MetNoFimex::Units::Units ()

initialization of unit handling, i.e. parsing of unit file etc if required

14.79.1.2 MetNoFimex::Units::Units (const Units & rhs)

14.79.1.3 virtual MetNoFimex::Units::~~Units () [virtual]

14.79.2 Member Function Documentation

14.79.2.1 Units& MetNoFimex::Units::operator= (const Units & rhs)

14.79.2.2 void MetNoFimex::Units::convert (const **std::string** & *from*, const **std::string** & *to*, double & *slope*, double & *offset*) throw (UnitException)

calculate the linear unit conversion: newVal (in to unit) = oldVal (in from unit) * slope + offset

Parameters:

from unit

to unit

slope return value of the slope

offset return value of the offset

14.79.2.3 bool MetNoFimex::Units::areConvertible (const **std::string** & *unit1*, const **std::string** & *unit2*) const throw (UnitException)

test if two units are convertible to each others

Parameters:

- unit1* first unit
- unit2* second unit

14.79.2.4 bool MetNoFimex::Units::isTime (const std::string & *timeUnit*) const throw (UnitException)

The documentation for this class was generated from the following file:

- include/fimex/[Units.h](#)

14.80 MetNoFimex::XMLDoc Class Reference

```
#include <XMLDoc.h>
```

Public Member Functions

- [XMLDoc](#) (const **std::string** &filename) throw (CDMException)
- virtual [~XMLDoc](#) ()
- [XPathObjPtr getXPathObject](#) (const **std::string** &xpath, [xmlNodePtr](#) node=0) const throw (CDMException)
- void [registerNamespace](#) (const **std::string** &prefix, const **std::string** &uri) throw (CDMException)

register a namespace for later xpath

14.80.1 Detailed Description

a tiny wrapper around libxml dom and xpath reader with xml::include

14.80.2 Constructor & Destructor Documentation

14.80.2.1 MetNoFimex::XMLDoc::XMLDoc (const **std::string** & *filename*) throw (CDMException)

initialization of libxml and the xml config file

Parameters:

filename xml input-file

Exceptions:

[CDMException](#) if problems with libxml or problems with input-file

14.80.2.2 virtual MetNoFimex::XMLDoc::~~XMLDoc () [virtual]

14.80.3 Member Function Documentation

14.80.3.1 XPathObjPtr MetNoFimex::XMLDoc::getXPathObject (const **std::string** & *xpath*, [xmlNodePtr](#) *node* = 0) const throw (CDMException)

get a ptr to the node defined by xpath

Parameters:

xpath xpath string for the node

Returns:

an xpathobj, which is != 0, but might have 0 elements, i.e. nodesetval == 0 or nodesetval->nodeNr == 0

Exceptions:

CDMException if xpath is not parsable

14.80.3.2 void MetNoFimex::XMLDoc::registerNamespace (const std::string & *prefix*, const std::string & *uri*) throw (CDMException)

register a namespace for later xpath

register a namespace with a prefix for later xpath retrievals

Parameters:

prefix short name for namespace

uri full namespace name

The documentation for this class was generated from the following file:

- include/fimex/[XMLDoc.h](#)

Chapter 15

File Documentation

15.1 doxydoc.txt File Reference

15.2 include/felt/FeltConstants.h File Reference

```
#include <algorithm>
#include <boost/static_assert.hpp>
```

Namespaces

- namespace [felt](#)

Defines

- #define [M_PI](#) 3.14159265358979323846

Typedefs

- typedef short int [felt::word](#)
A [felt](#) block "word" - 2 bytes.

Functions

- bool [felt::isUndefined](#) (word w)
- [BOOST_STATIC_ASSERT](#) (sizeof([felt::word](#))==2)

Variables

- const size_t [felt::blockWords](#) = 1024
- const size_t [felt::blockSize](#) = blockWords * sizeof(word)
- const size_t [felt::offsetToContentDefinition](#) = 2
- const double [felt::PI](#) = [M_PI](#)
- const double [felt::EARTH_RADIUS](#) = 6371000.

15.2.1 Define Documentation

15.2.1.1 #define [M_PI](#) 3.14159265358979323846

15.2.2 Function Documentation

15.2.2.1 [BOOST_STATIC_ASSERT](#) (sizeof([felt::word](#)) == 2)

15.3 include/felt/FeltField.h File Reference

```
#include "FeltConstants.h"
#include "FeltGridDefinition.h"
#include <boost/shared_array.hpp>
#include <boost/array.hpp>
#include <boost/date_time/posix_time/posix_time_types.hpp>
#include <boost/noncopyable.hpp>
#include <string>
#include <vector>
```

Namespaces

- namespace [felt](#)

Classes

- class [felt::FeltField](#)

Typedefs

- typedef boost::shared_ptr< FeltGridDefinition > [felt::FeltGridDefinitionPtr](#)

15.4 include/felt/FeltFile.h File Reference

```
#include "FeltConstants.h"
#include "FeltTypes.h"
#include <boost/shared_ptr.hpp>
#include <boost/weak_ptr.hpp>
#include <boost/filesystem/path.hpp>
#include <boost/date_time/posix_time/posix_time_types.hpp>
#include <boost/shared_array.hpp>
#include <boost/noncopyable.hpp>
#include <iterator>
#include <vector>
#include <iosfwd>
```

Namespaces

- namespace [felt](#)

Classes

- class [felt::FeltFile](#)

15.5 include/felt/FeltGridDefinition.h File Reference

```
#include <string>
#include <vector>
#include <iosfwd>
#include <boost/array.hpp>
```

Namespaces

- namespace [felt](#)

Classes

- class [felt::FeltGridDefinition](#)

Functions

- [boost::array< float, 6 > felt::gridParameters](#) (int gridType, int xNum, int yNum, int a, int b, int c, int d, const **std::vector**< short int > &extraData)
- **std::string** [felt::gridParametersToProjDefinition](#) (int gridType, const boost::array< float, 6 > &gridPars)
- **std::ostream** & [felt::contentSummary](#) (**std::ostream** &out, const FeltGridDefinition &grid)

15.6 include/felt/FeltTypeConversion.h File Reference

```
#include "FeltConstants.h"
#include <boost/date_time/posix_time/posix_time_types.hpp>
```

Namespaces

- namespace [felt](#)

Functions

- template<typename T>
T [felt::get](#) (word w)
- boost::posix_time::ptime [felt::parseTime](#) (const word *data)
- boost::posix_time::ptime [felt::parseTimeNoThrow](#) (const word *data)

15.7 include/felt/FeltTypes.h File Reference

Namespaces

- namespace [felt](#)

15.8 include/fimex/binaryConstants.h File Reference

Classes

- struct [binary< N >](#)
- struct [binary< 0 >](#)

15.9 include/fimex/C_CDMReader.h File Reference

```
#include "fimex/CDMReader.h"  
#include "fimex/c_fimex.h"  
#include <map>
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::C_CDMReader](#)

15.10 include/fimex/c_fimex.h File Reference

```
#include <stddef.h>
```

Typedefs

- typedef struct [mifi_cdm_reader](#) [mifi_cdm_reader](#)
- typedef int(* [doubleDatasliceCallbackPtr](#))([mifi_cdm_reader](#) *reader, const char *varName, size_t unLimDimPos, double *scaledData, size_t dataSize)

Functions

- void [mifi_free_cdm_reader](#) ([mifi_cdm_reader](#) *reader)
- [mifi_cdm_reader](#) * [mifi_new_felt_reader](#) (const char *filename, const char *configFile)
- [mifi_cdm_reader](#) * [mifi_new_netcdf_reader](#) (const char *filename)
- [mifi_cdm_reader](#) * [mifi_new_grib_reader](#) (const char *filename, const char *configFile)
- [mifi_cdm_reader](#) * [mifi_new_ncml_reader](#) (const char *ncmlFile)
- [mifi_cdm_reader](#) * [mifi_new_ncml_modifier](#) ([mifi_cdm_reader](#) *reader, const char *ncmlFile)
- int [mifi_netcdf_writer](#) ([mifi_cdm_reader](#) *reader, const char *filename, const char *configFile, int version)
- int [mifi_grib_writer](#) ([mifi_cdm_reader](#) *reader, const char *filename, const char *configFile, int version)
- int [mifi_nullcdm_writer](#) ([mifi_cdm_reader](#) *reader)
- [mifi_cdm_reader](#) * [mifi_new_cdminterpolator](#) ([mifi_cdm_reader](#) *reader, int method, const char *proj_input, const char *out_x_axis, const char *out_y_axis, const char *out_x_axis_unit, const char *out_y_axis_unit)
change the projection of the reader to this new projection
- [mifi_cdm_reader](#) * [mifi_new_c_reader](#) ([mifi_cdm_reader](#) *reader)
- int [mifi_set_callback_double](#) ([mifi_cdm_reader](#) *c_reader, const char *varName, [doubleDatasliceCallbackPtr](#) callback)
- size_t [mifi_get_variable_number](#) ([mifi_cdm_reader](#) *reader)
- const char * [mifi_get_variable_name](#) ([mifi_cdm_reader](#) *reader, size_t pos)
- int [mifi_get_double_dataslice](#) ([mifi_cdm_reader](#) *reader, const char *varName, size_t unLimDimPos, double **data, size_t *size)
- int [mifi_get_double_data](#) ([mifi_cdm_reader](#) *reader, const char *varName, double **data, size_t *size)

15.10.1 Typedef Documentation

15.10.1.1 typedef int(* [doubleDatasliceCallbackPtr](#))([mifi_cdm_reader](#) *reader, const char *varName, size_t unLimDimPos, double *scaledData, size_t dataSize)

Function pointer as used for the [get_double_dataslice](#) callback function

Returns:

0 on success, error otherwise

15.10.1.2 typedef struct mifi_cdm_reader mifi_cdm_reader

This is the public C-API for fimex. It is a wrapper api for the underlying C++ api.

15.10.2 Function Documentation

15.10.2.1 void mifi_free_cdm_reader (mifi_cdm_reader * reader)

Free the reader. This won't free the resources immediately, but reduce the reference counter. It is therefore possible to free a reader, while it still is used within another part of the fimex-chain.

15.10.2.2 int mifi_get_double_data (mifi_cdm_reader * reader, const char * varName, double ** data, size_t * size)

get all the data from the dataReader

Parameters:

reader dataReader to read the data from

varName variable name associated with the data

data,: the returned data. It will be allocated automatically, it is the task of the user to **free** it. Undefined values will be NaN.

size,: the size of the returned data.

Returns:

0 on success

15.10.2.3 int mifi_get_double_datasize (mifi_cdm_reader * reader, const char * varName, size_t unLimDimPos, double ** data, size_t * size)

get a slice of data from the dataReader

Parameters:

reader dataReader to read the data from

varName variable name associated with the data

unLimDimPos unlimited dimension of the slice

data,: the returned data. It will be allocated automatically, it is the task of the user to **free** it. Undefined values will be NaN.

size,: the size of the returned data.

Returns:

0 on success

15.10.2.4 `const char* mifi_get_variable_name (mifi_cdm_reader * reader, size_t pos)`

Get the name of a variable from the reader.

Parameters:

reader the data source

pos the position number of the variable, should be between 0 and size-1

Returns:

the variable name, or NULL on failure

15.10.2.5 `size_t mifi_get_variable_number (mifi_cdm_reader * reader)`

Get the number of the variables from the reader.

Parameters:

reader the data source

Returns:

the number of variables

15.10.2.6 `int mifi_grib_writer (mifi_cdm_reader * reader, const char * filename, const char * configFile, int version)`

Write the content of the reader to the filename as gribfile.

Parameters:

reader the data source

filename the name of the grib-file to write

configFile an optional configFile, use "" or 0 if not needed

version,the version of the grib-edition. Implemented are 1 or 2.

Returns:

0 on success.

15.10.2.7 `int mifi_netcdf_writer (mifi_cdm_reader * reader, const char * filename, const char * configFile, int version)`

Write the content of the reader to the filename.

Parameters:

reader the data source

filename the name of the netcdf-file to write

configFile an optional configFile, use "" or 0 if not needed

version,the version of the netcdf-file. Implemented are 3 or 4.

Returns:

0 on success.

15.10.2.8 mifi_cdm_reader* mifi_new_c_reader (mifi_cdm_reader * *reader*)

Get a new reader which allows setting c-callback functions.

Parameters:

the original data-source

Returns:

the reader object-pointer, use mifi_freeCDMReader to free, or NULL on error.

15.10.2.9 mifi_cdm_reader* mifi_new_cdminterpolator (mifi_cdm_reader * *reader*, int *method*, const char * *proj_input*, const char * *out_x_axis*, const char * *out_y_axis*, const char * *out_x_axis_unit*, const char * *out_y_axis_unit*)

change the projection of the reader to this new projection

Parameters:

method Interpolation method

proj_input input-string for proj4, used as output projection

out_x_axis config-string for x_axis, either '1,2,...,5' or 'auto' or 'auto,distance=3.5'

out_y_axis config-string for y_axis, either '1,2,...,5' or 'auto' or 'auto,distance=3.5'

out_x_axis_unit unit of the output x-axis

out_y_axis_unit unit of the output y-axis

Returns:

the reader object-pointer, use mifi_freeCDMReader to free, or NULL on error.

15.10.2.10 mifi_cdm_reader* mifi_new_felt_reader (const char * *filename*, const char * *configFile*)

Get a new reader from a [felt](#) file.

Parameters:

filename name of the felt-file

configFile configuration file for the felt-file

Returns:

the reader object-pointer, use mifi_freeCDMReader to free, or NULL on error.

15.10.2.11 mifi_cdm_reader* mifi_new_grib_reader (const char * *filename*, const char * *configFile*)

Get a new reader from a grib1/2 file.

Parameters:

filename name of the grib-file

configFile configuration file for the grib-file

Returns:

the reader object-pointer, use `mifi_freeCDMReader` to free, or NULL on error.

15.10.2.12 `mifi_cdm_reader* mifi_new_ncml_modifier (mifi_cdm_reader * reader, const char * ncmlFile)`

Modify a reader using a ncml file.

Parameters:

reader the data/cdm source

ncmlFile name of the ncml config file

Returns:

the reader object-pointer, use `mifi_freeCDMReader` to free, or NULL on error.

15.10.2.13 `mifi_cdm_reader* mifi_new_ncml_reader (const char * ncmlFile)`

Get a new reader from a ncml file.

Parameters:

ncmlFile name of the ncml config file

Returns:

the reader object-pointer, use `mifi_freeCDMReader` to free, or NULL on error.

15.10.2.14 `mifi_cdm_reader* mifi_new_netcdf_reader (const char * filename)`

Get a new reader from a netcdf file.

Parameters:

filename name of the felt-file

configFile configuration file for the felt-file

Returns:

the reader object-pointer, use `mifi_freeCDMReader` to free, or NULL on error.

15.10.2.15 `int mifi_nullcdm_writer (mifi_cdm_reader * reader)`

Fetch the whole data belonging to the cdm, but don't write it anywhere.

Parameters:

reader the data source

Returns:

0 on success.

15.10.2.16 `int mifi_set_callback_double (mifi_cdm_reader * c_reader, const char * varName, doubleDatasliceCallbackPtr callback)`

Add a callback for a variable. The variable will be converted to datatype double.

Parameters:

c_reader the reader as created by [mifi_new_c_reader](#)

varName the name of the variable

callback a function-ptr to the callback function

Returns:

0 on success, else error

Warning:

the callback function will only be able to modify data which is available in the reader. It cannot change any information the writer request, but the reader doesn't now about. This data will continue to be undefined!

15.11 `include/fimex/CachedForwardInterpolation.h` File Reference

```
#include "fimex/CachedInterpolation.h"  
#include <boost/shared_array.hpp>  
#include <vector>
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::CachedForwardInterpolation](#)

15.12 include/fimex/CachedInterpolation.h File Reference

```
#include <boost/shared_ptr.hpp>
#include <boost/shared_array.hpp>
#include "fimex/interpolation.h"
#include "fimex/Data.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::CachedInterpolationInterface](#)
- class [MetNoFimex::CachedInterpolation](#)

15.13 include/fimex/CachedVectorReprojection.h File Reference

```
#include <boost/shared_ptr.hpp>
#include "fimex/Data.h"
#include "fimex/interpolation.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::CachedVectorReprojection](#)

15.14 include/fimex/CDM.h File Reference

```
#include <map>
#include <vector>
#include <string>
#include <ostream>
#include <boost/regex_fwd.hpp>
#include "fimex/CDMAttribute.h"
#include "fimex/CDMVariable.h"
#include "fimex/CDMDimension.h"
#include "fimex/CDMException.h"
#include "fimex/CDMconstants.h"
#include "fimex/coordSys/Projection.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::CDM](#)
Data structure of the Common Data Model.

15.15 include/fimex/CDMAttribute.h File Reference

```
#include <string>
#include <vector>
#include <ostream>
#include <boost/shared_ptr.hpp>
#include "fimex/CDMDataType.h"
#include "fimex/CDMNamedEntity.h"
#include "fimex/CDMException.h"
#include "fimex/deprecated.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::CDMAttribute](#)

Functions

- [MetNoFimex::DEPRECATED](#) (**std::vector**< CDMAttribute > projStringToAttributes(**std::string** projStr))
convert a proj4 string to a list of CDMAttributes usable for CF-1.0 projection variable
- [MetNoFimex::DEPRECATED](#) (**std::string** attributesToProjString(const **std::vector**< CDMAttribute > &attrs))
convert attributes of a projection-variable to a projString

15.16 include/fimex/CDMconstants.h File Reference

Defines

- #define [MIFI_EARTH_RADIUS_M](#) 6371000
constants used through-out fimex
- #define [MIFI_FILETYPE_UNKNOWN](#) -1
- #define [MIFI_FILETYPE_FELT](#) 0
- #define [MIFI_FILETYPE_NETCDF](#) 1
- #define [MIFI_FILETYPE_NCML](#) 2
- #define [MIFI_FILETYPE_GRIB](#) 3

Functions

- const char * [fimexVersion](#) ()
- int [fimexHasNetcdf](#) ()
- int [fimexHasGribApi](#) ()
- int [fimexHasFelt](#) ()

15.16.1 Define Documentation

15.16.1.1 #define MIFI_EARTH_RADIUS_M 6371000

constants used through-out fimex

CDMConstants stores several constants used in fimex, accessible from C and C++. Constants are either available as macro, or as function. the default radius of a sperical earth in meter

15.16.1.2 #define MIFI_FILETYPE_FELT 0

15.16.1.3 #define MIFI_FILETYPE_GRIB 3

15.16.1.4 #define MIFI_FILETYPE_NCML 2

15.16.1.5 #define MIFI_FILETYPE_NETCDF 1

15.16.1.6 #define MIFI_FILETYPE_UNKNOWN -1

The MIFI_FILETYPE_* define the available input and output file-formats

15.16.2 Function Documentation

15.16.2.1 int fimexHasFelt ()

check if fimex is configured with felt-support

15.16.2.2 int fimexHasGribApi ()

check if fimex is configured with grib_api-support

15.16.2.3 int fimexHasNetcdf ()

check if fimex is configured with netcdf-support

15.16.2.4 const char* fimexVersion ()

version of fimex

15.17 include/fimex/CDMDataType.h File Reference

```
#include <string>
```

Namespaces

- namespace [MetNoFimex](#)

Enumerations

- enum [MetNoFimex::CDMDataType](#) {
 [MetNoFimex::CDM_NAT](#) = 0, [MetNoFimex::CDM_CHAR](#), [MetNoFimex::CDM_SHORT](#),
 [MetNoFimex::CDM_INT](#),
 [MetNoFimex::CDM_FLOAT](#), [MetNoFimex::CDM_DOUBLE](#), [MetNoFimex::CDM_STRING](#) }

Functions

- CDMDataType [MetNoFimex::string2datatype](#) (const **std::string** &s)
 translate float/string/... to the appropriate CDMDataType
- **std::string** [MetNoFimex::datatype2string](#) (CDMDataType type)

15.18 include/fimex/CDMDimension.h File Reference

```
#include <string>
#include <ostream>
#include "fimex/CDMNamedEntity.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::CDMDimension](#)

15.19 include/fimex/CDMException.h File Reference

```
#include <stdexcept>
#include <string>
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::CDMException](#)

15.20 include/fimex/CDMExtractor.h File Reference

```
#include <map>
#include <boost/array.hpp>
#include <boost/shared_ptr.hpp>
#include "fimex/CDMReader.h"
#include "fimex/CDMDataType.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::CDMExtractor](#)

15.21 include/fimex/CDMFileReaderFactory.h File Reference

```
#include <boost/shared_ptr.hpp>
#include <vector>
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::CDMFileReaderFactory](#)

15.22 include/fimex/CDMInterpolator.h File Reference

```
#include <vector>
#include "fimex/CDMReader.h"
#include "fimex/CachedInterpolation.h"
#include "fimex/CachedVectorReprojection.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::CDMInterpolator](#)

15.23 include/fimex/CDMNamedEntity.h File Reference

```
#include <string>
#include <functional>
#include <boost/shared_ptr.hpp>
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::CDMNamedEntity](#)
- struct [MetNoFimex::CDMNameCompare](#)
- class [MetNoFimex::CDMNameEqual](#)
- class [MetNoFimex::CDMNameEqualPtr](#)

15.24 include/fimex/CDMQualityExtractor.h File Reference

```
#include "CDMReader.h"
#include <vector>
#include <map>
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::CDMQualityExtractor](#)
Extract data with defined quality status.

15.25 include/fimex/CDMReader.h File Reference

```
#include <boost/shared_ptr.hpp>
#include "fimex/CDMException.h"
#include "fimex/SliceBuilder.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::CDMReader](#)
Basic interface for [CDM](#) reading and manipulation classes.

15.26 `include/fimex/CDMTimeInterpolator.h` File Reference

```
#include "CDMReader.h"  
#include <map>  
#include <vector>
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::CDMTimeInterpolator](#)

15.27 include/fimex/CDMVariable.h File Reference

```
#include <string>
#include <vector>
#include <ostream>
#include "fimex/CDMAttribute.h"
#include "fimex/CDMDataType.h"
#include "fimex/CDMNamedEntity.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::CDMVariable](#)

15.28 include/fimex/CDMWriter.h File Reference

```
#include <string>
#include <boost/shared_ptr.hpp>
#include "fimex/CDMReader.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::CDMWriter](#)

15.29 include/fimex/coordSys/CoordinateAxis.h File Reference

```
#include "fimex/CDMVariable.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::CoordinateAxis](#)

Functions

- `std::ostream & MetNoFimex::operator<< (std::ostream &out, CoordinateAxis ca)`
- `std::ostream & MetNoFimex::operator<< (std::ostream &out, CoordinateAxis::AxisType t)`

15.30 include/fimex/coordSys/CoordinateSystem.h File Reference

```
#include <functional>
#include <vector>
#include <boost/shared_ptr.hpp>
#include <iostream>
#include "fimex/coordSys/CoordinateAxis.h"
#include "fimex/coordSys/Projection.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::CoordinateSystem](#)
- struct [MetNoFimex::CompleteCoordinateSystemForComparator](#)

Functions

- **std::ostream** & [MetNoFimex::operator<<](#) (**std::ostream** &out, const CoordinateSystem &p)
- **std::vector**< boost::shared_ptr< const CoordinateSystem > > [MetNoFimex::listCoordinateSystems](#) (const CDM &cdm)

15.31 include/fimex/coordSys/LambertConformalConicProjection.h File Reference

```
#include "fimex/coordSys/ProjectionImpl.h"  
#include "fimex/Data.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::LambertConformalConicProjection](#)

15.32 `include/fimex/coordSys/LatitudeLongitudeProjection.h` File Reference

```
#include "fimex/coordSys/ProjectionImpl.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::LatitudeLongitudeProjection](#)

15.33 include/fimex/coordSys/MercatorProjection.h File Reference

```
#include "fimex/coordSys/ProjectionImpl.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::MercatorProjection](#)

15.34 `include/fimex/coordSys/PolarStereographicProjection.h` File Reference

```
#include "fimex/coordSys/StereographicProjection.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::PolarStereographicProjection](#)

15.35 include/fimex/coordSys/Projection.h File Reference

```
#include <vector>
#include <iostream>
#include <boost/shared_ptr.hpp>
#include "fimex/CDMAAttribute.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::Projection](#)

Functions

- `std::ostream & MetNoFimex::operator<< (std::ostream &out, const Projection &proj)`

15.36 `include/fimex/coordSys/ProjectionImpl.h` File Reference

```
#include "fimex/coordSys/Projection.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::ProjectionImpl](#)

15.37 include/fimex/coordSys/RotatedLatitudeLongitudeProjection.h File Reference

```
#include "fimex/coordSys/ProjectionImpl.h"  
#include "fimex/Data.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::RotatedLatitudeLongitudeProjection](#)

15.38 `include/fimex/coordSys/StereographicProjection.h` File Reference

```
#include "fimex/coordSys/ProjectionImpl.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::StereographicProjection](#)

15.39 include/fimex/Data.h File Reference

```
#include <boost/shared_array.hpp>
#include <boost/shared_ptr.hpp>
#include <string>
#include <sstream>
#include <iostream>
#include "fimex/CDMDataType.h"
#include "fimex/CDMException.h"
#include "fimex/Utils.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::Data](#)

Functions

- `boost::shared_ptr< Data > MetNoFimex::createData (CDMDataType datatype, size_t length, double val=0) throw (CDMException)`
create a Data-pointer of the datatype
- `boost::shared_ptr< Data > MetNoFimex::createDataSlice (CDMDataType datatype, const Data &data, size_t dataStartPos, size_t dataSize) throw (CDMException)`
create a one-dimensional dataslice from another [Data](#) object

15.40 include/fimex/DataImpl.h File Reference

```
#include <typeinfo>
#include <boost/shared_ptr.hpp>
#include <string>
#include <sstream>
#include <iostream>
#include <iterator>
#include <cmath>
#include <algorithm>
#include "fimex/Data.h"
#include "fimex/CDMDataType.h"
#include "fimex/CDMException.h"
#include "fimex/Utils.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::DataImpl< C >](#)

Functions

- `template<typename T1, typename T2>`
`boost::shared_array< T1 > MetNoFimex::duplicateArrayType (const boost::shared_array< T2 > &inData, long length)`
create a new shared array with a different type using static_cast
- `template<typename T1, typename T2>`
`const boost::shared_array< T1 > MetNoFimex::constConvertArrayType (const boost::shared_array< T2 > &inData, long length)`
return a shared array of this data, possibly pointer to internal data
- `template<class InputIterator>`
`boost::shared_ptr< Data > MetNoFimex::createData (CDMDataType datatype, InputIterator first, InputIterator last) throw (CDMException)`
create a Data-pointer of the datatype and fill with the data from the iterator
- `template<typename C>`
`void MetNoFimex::recursiveCopyMultiDimData (C **orgData, C **newData, const std::vector< size_t > &orgDimSize, const std::vector< size_t > &orgSliceSize, const std::vector< size_t > &newStart, const std::vector< size_t > &newSize, size_t currentDim)`

- `template<typename OUT, typename IN>`
`boost::shared_array< OUT > MetNoFimex::convertArrayType (const boost::shared_array< IN >`
`&inData, size_t length, double oldFill, double oldScale, double oldOffset, double newFill, double`
`newScale, double newOffset)`

15.41 `include/fimex/DataTypeChanger.h` File Reference

```
#include "fimex/CDMDataType.h"  
#include "boost/shared_ptr.hpp"  
#include "fimex/CDMException.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::DataTypeChanger](#)

15.42 include/fimex/deprecated.h File Reference

Defines

- #define [DEPRECATED](#)(func) func

15.42.1 Define Documentation

15.42.1.1 #define DEPRECATED(func) func

15.43 include/fimex/Felt_Array.h File Reference

```
#include <string>
#include <set>
#include <vector>
#include <map>
#include <boost/array.hpp>
#include "fimex/Felt_File_Error.h"
#include "fimex/Utils.h"
```

Namespaces

- namespace [MetNoFelt](#)

Classes

- struct [MetNoFelt::ShortPairLess](#)
- class [MetNoFelt::Felt_Array](#)
encapsulate parameters of a [felt](#) file

Typedefs

- typedef set< pair< short, short >, ShortPairLess > [MetNoFelt::ShortPairSet](#)
- typedef map< pair< short, short >, short, ShortPairLess > [MetNoFelt::ShortPairMap](#)

Functions

- epoch_seconds [MetNoFelt::index16toTime](#) (const boost::array< short, 16 > &idx)
- pair< short, short > [MetNoFelt::index16toLevelPair](#) (const boost::array< short, 16 > &idx)

15.44 include/fimex/Felt_Array2.h File Reference

```
#include <string>
#include <set>
#include <vector>
#include <map>
#include <boost/array.hpp>
#include <boost/date_time/posix_time/posix_time_types.hpp>
#include <fimex/Felt_Types.h>
#include "fimex/Felt_File_Error.h"
#include "felt/FeltTypes.h"
```

Namespaces

- namespace [MetNoFelt](#)

Classes

- class [MetNoFelt::Felt_Array2](#)

A collection of FeltFields to build a 4-dimensional data array for 1-parameter and 1-vertical coordinate.

15.45 include/fimex/Felt_File.h File Reference

```
#include <map>
#include <vector>
#include <string>
#include <boost/shared_ptr.hpp>
#include <boost/shared_array.hpp>
#include "fimex/Data.h"
#include "fimex/Felt_Array.h"
#include "fimex/Felt_File_Error.h"
#include "fimex/FeltParameters.h"
#include "fimex/Logger.h"
#include "fimex/Utils.h"
```

Namespaces

- namespace [MetNoFelt](#)

Classes

- class [MetNoFelt::Felt_File](#)
Felt File access.

15.46 include/fimex/Felt_File2.h File Reference

```
#include <map>
#include <vector>
#include <string>
#include <boost/shared_ptr.hpp>
#include <boost/shared_array.hpp>
#include "fimex/Data.h"
#include "fimex/Felt_Types.h"
#include "fimex/Felt_File_Error.h"
#include "fimex/FeltParameters.h"
#include "fimex/Logger.h"
#include "felt/FeltTypes.h"
```

Namespaces

- namespace [MetNoFelt](#)

Classes

- class [MetNoFelt::Felt_File2](#)
Felt File access.

15.47 `include/fimex/Felt_File_Error.h` File Reference

```
#include <stdexcept>
#include <string>
```

Namespaces

- namespace [MetNoFelt](#)

Classes

- class [MetNoFelt::Felt_File_Error](#)

15.48 include/fimex/Felt_Types.h File Reference

Namespaces

- namespace [MetNoFelt](#)

Classes

- struct [MetNoFelt::LevelPairLess](#)

Typedefs

- typedef `std::pair< short, short >` [MetNoFelt::LevelPair](#)

15.49 include/fimex/FeltCDMReader.h File Reference

```
#include <string>
#include <vector>
#include <map>
#include <boost/shared_ptr.hpp>
#include "fimex/Felt_File.h"
#include "fimex/CDMReader.h"
#include "fimex/CDMDimension.h"
#include "fimex/ReplaceStringObject.h"
#include "fimex/Utils.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::FeltCDMReader](#)

15.50 include/fimex/FeltCDMReader2.h File Reference

```
#include <string>
#include <vector>
#include <map>
#include <boost/shared_ptr.hpp>
#include "fimex/CDMReader.h"
#include "fimex/CDMDimension.h"
#include "fimex/Felt_Types.h"
#include "fimex/Felt_File_Error.h"
#include "fimex/ReplaceStringObject.h"
#include <boost/date_time/posix_time/posix_time_types.hpp>
```

Namespaces

- namespace [MetNoFelt](#)
- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::FeltCDMReader2](#)

Functions

- `boost::shared_ptr< FeltCDMReader2 > MetNoFimex::getFeltReader (std::string filename, std::string configFilename)`
convenience function to retrieve a reader as shared_ptr instead of calling new

15.51 include/fimex/FeltParameters.h File Reference

```
#include <map>
#include <string>
#include <vector>
#include <boost/array.hpp>
#include "fimex/Felt_File_Error.h"
```

Namespaces

- namespace [MetNoFelt](#)

Classes

- class [MetNoFelt::FeltParameters](#)

Functions

- **std::string** [MetNoFelt::getProjString](#) (int gridType, const boost::array< float, 6 > &gridParameters) throw (Felt_File_Error)
- const int [MetNoFelt::ANY_VALUE](#) ()
- const **std::string** & [MetNoFelt::UNDEFINED](#) ()
- const boost::array< short, 16 > & [MetNoFelt::ANY_ARRAY](#) ()
- const boost::array< short, 20 > & [MetNoFelt::ANY_ARRAY20](#) ()

15.52 include/fimex/GribApiCDMWriter.h File Reference

```
#include "fimex/CDMWriter.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::GribApiCDMWriter](#)

15.53 include/fimex/GribApiCDMWriter_Impl1.h File Reference

```
#include "fimex/GribApiCDMWriter_ImplAbstract.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::GribApiCDMWriter_Impl1](#)

15.54 include/fimex/GribApiCDMWriter_Impl2.h File Reference

```
#include "fimex/GribApiCDMWriter_ImplAbstract.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::GribApiCDMWriter_Impl2](#)

15.55 include/fimex/GribApiCDMWriter_ImplAbstract.h File Reference

```
#include <vector>
#include <grib_api.h>
#include <fstream>
#include <iostream>
#include "fimex/Logger.h"
#include "fimex/CDMWriter.h"
#include "fimex/XMLDoc.h"
#include "fimex/CDMException.h"
#include "fimex/TimeUnit.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::GribApiCDMWriter_ImplAbstract](#)

15.56 include/fimex/GribCDMReader.h File Reference

```
#include <vector>
#include <map>
#include <set>
#include "boost/shared_ptr.hpp"
#include "fimex/GribFileIndex.h"
#include "fimex/CDMReader.h"
#include "fimex/ReplaceStringObject.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::GribCDMReader](#)

15.57 include/fimex/GribFileIndex.h File Reference

```
#include <boost/date_time/posix_time/posix_time.hpp>
#include <boost/filesystem.hpp>
#include <vector>
#include "fimex/XMLDoc.h"
#include "fimex/GridDefinition.h"
#include <boost/date_time/posix_time/posix_time_types.hpp>
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::GribFileMessage](#)
- class [MetNoFimex::GribFileMessageEqualTime](#)
Functor to find Messages with equal time.
- class [MetNoFimex::GribFileMessageEqualLevelTime](#)
Functor to find messages with equal level and time.
- class [MetNoFimex::GribFileIndex](#)

Functions

- `size_t MetNoFimex::gribDataRead (const GribFileMessage &gfm, std::vector< double > &data, double missingValue)`
- `std::ostream & MetNoFimex::operator<< (std::ostream &os, const GribFileMessage &gfm)`
outputstream for a [GribFileMessage](#)
- `std::ostream & MetNoFimex::operator<< (std::ostream &os, const GribFileIndex &gfm)`
outputstream for a [GribFileIndex](#)

15.58 include/fimex/GribUtils.h File Reference

```
#include "fimex/GridDefinition.h"
#include <stdexcept>
```

Namespaces

- namespace [MetNoFimex](#)

Defines

- #define [MIFI_GRIB_CHECK](#)(error, msg) mifi_grib_check(error, msg, __LINE__, __FILE__);

Functions

- void [mifi_grib_check](#) (int error, const char *msg, int line, const char *file) throw (std::runtime_error)
- GridDefinition::Orientation [MetNoFimex::gribGetGridOrientation](#) (boost::shared_ptr< grib_handle > gh)

15.58.1 Define Documentation

15.58.1.1 #define [MIFI_GRIB_CHECK](#)(error, msg) mifi_grib_check(error, msg, __LINE__, __FILE__);

macro to call [mifi_grib_check](#) with correct line and file

15.58.2 Function Documentation

15.58.2.1 void [mifi_grib_check](#) (int *error*, const char * *msg*, int *line*, const char * *file*) throw (std::runtime_error)

runtime-exception checker for grib_check

15.59 include/fimex/GridDefinition.h File Reference

```
#include "fimex/binaryConstants.h"
#include <string>
#include "boost/shared_ptr.hpp"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::GridDefinition](#)

15.60 include/fimex/interpolation.h File Reference

```
#include <proj_api.h>
#include <math.h>
```

Defines

- #define [PI](#) 3.1415926535897932384626433832795
- #define [MIFI_INTERPOL_NEAREST_NEIGHBOR](#) 0
interpolation method
- #define [MIFI_INTERPOL_BILINEAR](#) 1
interpolation method
- #define [MIFI_INTERPOL_BICUBIC](#) 2
interpolation method
- #define [MIFI_INTERPOL_COORD_NN](#) 3
interpolation method
- #define [MIFI_INTERPOL_COORD_NN_KD](#) 4
interpolation method
- #define [MIFI_INTERPOL_FORWARD_SUM](#) 5
interpolation method
- #define [MIFI_INTERPOL_FORWARD_MEAN](#) 6
interpolation method
- #define [MIFI_INTERPOL_FORWARD_MEDIAN](#) 7
interpolation method
- #define [MIFI_INTERPOL_FORWARD_MAX](#) 8
interpolation method
- #define [MIFI_INTERPOL_FORWARD_MIN](#) 9
interpolation method
- #define [MIFI_VECTOR_KEEP_SIZE](#) 0
vector projection flag
- #define [MIFI_VECTOR_RESIZE](#) 1
vector projection flag
- #define [MIFI_UNDEFINED_F](#) (nan(""))
undefined value for floats
- #define [MIFI_UNDEFINED_D](#) (nan(""))
undefined value for doubles

- `#define MIFI_ERROR -1`
return code, error
- `#define MIFI_OK 1`
return code, ok
- `#define MIFI_PROJ_AXIS 0`
projection axis in m-equivalent
- `#define MIFI_LONGITUDE 1`
longitude projection axis in degrees
- `#define MIFI_LATITUDE 2`
latitude projection axis in degrees
- `#define MIFI_DEBUG 0`
debug flag

Functions

- `int mifi_interpolate_f` (int method, const char *proj_input, const float *infield, const double *in_x_axis, const double *in_y_axis, const int in_x_axis_type, const int in_y_axis_type, const int ix, const int iy, const int iz, const char *proj_output, float *outfield, const double *out_x_axis, const double *out_y_axis, const int out_x_axis_type, const int out_y_axis_type, const int ox, const int oy)
not implemented yet
- `int mifi_interpolate_d` (int method, char *proj_input, double *infield, double *in_x_axis, double *in_y_axis, int in_x_axis_type, int in_y_axis_type, int ix, int iy, int iz, char *proj_output, double *outfield, double *out_x_axis, double *out_y_axis, int out_x_axis_type, int out_y_axis_type, int ox, int oy)
not implemented yet
- `int mifi_vector_reproject_values_f` (int method, const char *proj_input, const char *proj_output, float *u_out, float *v_out, const double *out_x_axis, const double *out_y_axis, int out_x_axis_type, int out_y_axis_type, int ox, int oy, int oz)
interpolate the vector values
- `int mifi_vector_reproject_values_by_matrix_f` (int method, const double *matrix, float *u_out, float *v_out, int ox, int oy, int oz)
- `int mifi_get_vector_reproject_matrix` (const char *proj_input, const char *proj_output, const double *out_x_axis, const double *out_y_axis, int out_x_axis_type, int out_y_axis_type, int ox, int oy, double *matrix)
- `int mifi_get_values_f` (const float *infield, float *outfield, const double x, const double y, const int ix, const int iy, const int iz)
- `int mifi_get_values_bilinear_f` (const float *infield, float *outvalues, const double x, const double y, const int ix, const int iy, const int iz)
- `int mifi_get_values_bicubic_f` (const float *infield, float *outvalues, const double x, const double y, const int ix, const int iy, const int iz)
not implemented yet

- void [mifi_get_values_linear_f](#) (const float *infieldA, const float *infieldB, float *outfield, const size_t n, const double a, const double b, const double x)
- int [mifi_points2position](#) (double *points, const int n, const double *axis, const int num, const int axis_type)
find position in array of position in projection
- int [mifi_3d_array_position](#) (int x, int y, int z, int ix, int iy, int iz)
- int [mifi_project_values](#) (const char *proj_input, const char *proj_output, double *in_out_x_vals, double *in_out_y_vals, const int num)
project values so that the projection (x,y) => (x_proj), (y_proj) can be expressed as x_proj(x,y), y_proj(x,y)
- int [mifi_project_axes](#) (const char *proj_input, const char *proj_output, const double *in_x_axis, const double *in_y_axis, const int ix, const int iy, double *out_xproj_axis, double *out_yproj_axis)
project axes so that the projection (x,y) => (x_proj), (y_proj) can be expressed as x_proj(x,y), y_proj(x,y)
- size_t [mifi_bad2nanf](#) (float *posPtr, float *endPtr, float badVal)
- size_t [mifi_nanf2bad](#) (float *posPtr, float *endPtr, float badVal)

15.60.1 Define Documentation

15.60.1.1 #define MIFI_DEBUG 0

debug flag

15.60.1.2 #define MIFI_ERROR -1

return code, error

15.60.1.3 #define MIFI_INTERPOL_BICUBIC 2

interpolation method

flag for bicubic interpolation

15.60.1.4 #define MIFI_INTERPOL_BILINEAR 1

interpolation method

flag for bilinear interpolation

15.60.1.5 #define MIFI_INTERPOL_COORD_NN 3

interpolation method

Flag for nearest neighbor interpolation using lon/lat coordinates rather than the input projection. This is largely a brute force method which may take long time.

Vector projection is not implemented (not defined?)

Warning:

this works only from CDMInterpolator

15.60.1.6 #define MIFI_INTERPOL_COORD_NN_KD 4

interpolation method

Flag for nearest neighbor interpolation using coordinates with KD-tree. This works as nearest neighbor in the output-projection and has therefore numerical problems in some points, i.e. near southpole when using northpole-polarstereographic.

It doesn't work with output projections in degree, i.e. rotated latitude longitude.

Vector projection is not implemented (not defined?)

Warning:

this works only from CDMInterpolator

15.60.1.7 #define MIFI_INTERPOL_FORWARD_MAX 8

interpolation method

forward interpolation, maximum over all matching defined input-cells

15.60.1.8 #define MIFI_INTERPOL_FORWARD_MEAN 6

interpolation method

forward interpolation, averaging (mean) over all matching defined input-cells

15.60.1.9 #define MIFI_INTERPOL_FORWARD_MEDIAN 7

interpolation method

forward interpolation, median over all matching defined input-cells

15.60.1.10 #define MIFI_INTERPOL_FORWARD_MIN 9

interpolation method

forward interpolation, minimum over all matching defined input-cells

15.60.1.11 #define MIFI_INTERPOL_FORWARD_SUM 5

interpolation method

forward interpolation, summing over all matching input-cells

15.60.1.12 #define MIFI_INTERPOL_NEAREST_NEIGHBOR 0

interpolation method

flag for nearest neighbor interpolation

15.60.1.13 #define MIFI_LATITUDE 2

latitude projection axis in degrees

15.60.1.14 #define MIFI_LONGITUDE 1

longitude projection axis in degrees

15.60.1.15 #define MIFI_OK 1

return code, ok

15.60.1.16 #define MIFI_PROJ_AXIS 0

projection axis in m-equivalent

15.60.1.17 #define MIFI_UNDEFINED_D (nan(""))

undefined value for doubles

15.60.1.18 #define MIFI_UNDEFINED_F (nanf(""))

undefined value for floats

15.60.1.19 #define MIFI_VECTOR_KEEP_SIZE 0

vector projection flag

new size will be like old size

15.60.1.20 #define MIFI_VECTOR_RESIZE 1

vector projection flag

vector might change size with projection

15.60.1.21 #define PI 3.1415926535897932384626433832795**15.60.2 Function Documentation****15.60.2.1 int mifi_3d_array_position (int x, int y, int z, int ix, int iy, int iz)**

gives the position of an fortran like array of size ix, iy, iz

Returns:

the position of x, y, z

15.60.2.2 size_t mifi_bad2nanf (float * posPtr, float * endPtr, float badVal)

Convert bad-values to nan. The mifi_ functions don't handle bad values generally, but forward this work to the floating-point IEEE NaN's. This function converts a general bad value to a nan in a float array.

Parameters:

posPtr start pointer of the float array

endPtr end-pointer of the float array (excluded from conversion)

badVal bad value to be converted to nan

Returns:

number of conversions

15.60.2.3 int mifi_get_values_bicubic_f (const float * infield, float * outvalues, const double x, const double y, const int ix, const int iy, const int iz)

not implemented yet

The bicubic convolution algorithm assigns a value $f(x,y) = X * M * F * Mt * Yt$ with x, y between $(0 \leq x < 1)$, $X = (1, x, x^2, x^3)$, $Y = (1, y, y^2, y^3)$ and F a 4*4 matrix consisting of the original values of $f(-1,-1)$ to $f(2,2)$.

M is the convolution matrix with $a = -0.5$ as described by wikipedia (or Catmull-Rom for $a = 1$, not used here)

Mt and Yt are the transposed matrices/vector.

See also:

http://en.wikipedia.org/wiki/Bicubic_interpolation

<http://java.sun.com/products/java-media/jai/forDevelopers/jai-apidocs/javafx/media/>

15.60.2.4 int mifi_get_values_bilinear_f (const float * infield, float * outvalues, const double x, const double y, const int ix, const int iy, const int iz)

Bilinear interpolation requires a neighborhood extending one pixel to the right and below the central sample. If the fractional subsample position is given by (xfrac, yfrac), the resampled pixel value will be:

$$(1 - yfrac) * [(1 - xfrac) * s00 + xfrac * s01] + yfrac * [(1 - xfrac) * s10 + xfrac * s11]$$

This is documented by the following diagram:

$$\begin{array}{cc} s00 & s01 \\ & . < yfrac \end{array}$$

```

s10      s11
  ^
  xfrac

```

See also:

<http://java.sun.com/products/java-media/jai/forDevelopers/jai-apidocs/javafx/media/>

Warning:

if any of the 4 used values of *infield* is undefined or outside of *infield*, the return value will be undefined

15.60.2.5 `int mifi_get_values_f (const float * infield, float * outfield, const double x, const double y, const int ix, const int iy, const int iz)`

Get the nearest neighbor of a value. Values are rounded to array-position.

Parameters:

infield 3d fortran array of size *ix*,*iy*,*iz*

outfield 1d array of size *iz* containing the values

15.60.2.6 `void mifi_get_values_linear_f (const float * infieldA, const float * infieldB, float * outfield, const size_t n, const double a, const double b, const double x)`

Linear interpolation/extrapolation of values in the arrays *infieldA* and *infieldB* at position *a* and *b* to a field at *outfield* at position *x* with $o(x) = in(a) + (x - a) * (in(b) - in(a)) / (b - a)$

This interpolation can be used for linear time-interpolation.

Parameters:

infieldA array of size *n* with values of input at position *a*

infieldB array of size *n* with values of input at position *b*

outfield array of size *n* with values of input at position *x*, output

n size of arrays

a position of *infieldA*

b position of *infieldB*

x position of *outfield*

15.60.2.7 `int mifi_get_vector_reproject_matrix (const char * proj_input, const char * proj_output, const double * out_x_axis, const double * out_y_axis, int out_x_axis_type, int out_y_axis_type, int ox, int oy, double * matrix)`

calculate the vector reprojecton matrix used in [mifi_vector_reproject_values_f](#)

Parameters:

method (one of MIFI_VECTOR_KEEP_SIZE, MIFI_VECTOR_RESIZE)

proj_input proj4-string of projection of infield
proj_output proj4-string of projection of outfield
out_x_axis field of size ox. Axis needs to be strong monotonous and if longitude/latitude in degree
out_y_axis field of size oy. Axis needs to be strong monotonous and if longitude/latitude in degree
out_x_axis_type one of MIFI_LATITUDE, MIFI_LONGITUDE, MIFI_PROJ_AXIS
out_y_axis_type one of MIFI_LATITUDE, MIFI_LONGITUDE, MIFI_PROJ_AXIS
ox x-dimension of outfield
oy y-dimension of outfield
matrix matrix of size (4*ox*oy)

Returns:

MIFI_OK or error value

15.60.2.8 `int mifi_interpolate_d (int method, char *proj_input, double *infield, double *in_x_axis, double *in_y_axis, int in_x_axis_type, int in_y_axis_type, int ix, int iy, int iz, char *proj_output, double *outfield, double *out_x_axis, double *out_y_axis, int out_x_axis_type, int out_y_axis_type, int ox, int oy)`

not implemented yet

double version of mifi_interpolate_f

See also:

[mifi_interpolate_f](#)

15.60.2.9 `int mifi_interpolate_f (int method, const char *proj_input, const float *infield, const double *in_x_axis, const double *in_y_axis, const int in_x_axis_type, const int in_y_axis_type, const int ix, const int iy, const int iz, const char *proj_output, float *outfield, const double *out_x_axis, const double *out_y_axis, const int out_x_axis_type, const int out_y_axis_type, const int ox, const int oy)`

Interpolation between two projections. Missing values are set to MIFI_UNDEFINED_F which is implemented as C99 nanf. The coordinates of a cell give the midpoint of a cell, i.e. cell (10,20) spans ([9.5..10.5],[19.5-20.5])

Parameters:

method one of MIFI_INTERPOL_NEAREST_NEIGHBOR MIFI_INTERPOL_BILINEAR MIFI_INTERPOL_BICUBIC
proj_input proj4-string of projection of infield
infield real rectangular array of dimension infield[iz,iy,ix]
in_x_axis field of size ix. Axis needs to be strong monotonous and if longitude/latitude in degree
in_y_axis field of size iy. Axis needs to be strong monotonous and if longitude/latitude in degree
in_x_axis_type one of MIFI_LATITUDE, MIFI_LONGITUDE, MIFI_PROJ_AXIS
in_y_axis_type one of MIFI_LATITUDE, MIFI_LONGITUDE, MIFI_PROJ_AXIS
ix x-dimension of infield

iy y-dimension of infield

iz z-dimension of infield and outfield. The z-dim allows you to convert several fields at once without calculating the projection again and again.

proj_output proj4-string of projection of outfield

outfield real rectangular array of dimension outfield[iz,oy,ox]

out_x_axis field of size ox. Axis needs to be strong monotonous and if longitude/latitude in degree

out_y_axis field of size oy. Axis needs to be strong monotonous and if longitude/latitude in degree

out_x_axis_type one of MIFI_LATITUDE, MIFI_LONGITUDE, MIFI_PROJ_AXIS

out_y_axis_type one of MIFI_LATITUDE, MIFI_LONGITUDE, MIFI_PROJ_AXIS

ox x-dimension of outfield

oy y-dimension of outfield

15.60.2.10 `size_t mifi_nanf2bad (float * posPtr, float * endPtr, float badVal)`

Convert nan back to bad-values. See [mifi_bad2nanf](#)

Parameters:

posPtr start pointer of the float array

endPtr end-pointer of the float array (excluded from conversion)

badVal value NaNs will be converted to

Returns:

number of conversions

15.60.2.11 `int mifi_points2position (double * points, const int n, const double * axis, const int num, const int axis_type)`

find position in array of position in projection

points2position uses linear splines to find the array-position of points in the given axis

Parameters:

points the values will get changed from points in axis coordinates to array coordinates

n number of values in points

axis coordinate axis

num number of elements in coordinate axis

axis_type type of axis, one of MIFI_LONGITUDE, MIFI_LATITUDE, MIFI_PROJ_AXIS

15.60.2.12 `int mifi_project_axes (const char * proj_input, const char * proj_output, const double * in_x_axis, const double * in_y_axis, const int ix, const int iy, double * out_xproj_axis, double * out_yproj_axis)`

project axes so that the projection (x,y) => (x_proj), (y_proj) can be expressed as x_proj(x,y), y_proj(x,y)

all axes must be given or will be returned in radians when converted from/to latlon

Parameters:

proj_input input projection proj string
proj_output output projection proj string
in_x_axis x-axis in input-projection
in_y_axis y-axis in input-projection
ix size of x-axis
iy size of y-axis
out_xproj_axis output-values of $x_{proj}(x,y)$, field needs to be allocated in at least $ix*iy$ size
out_yproj_axis output-values of $y_{proj}(x,y)$, field needs to be allocated in at least $ix*iy$ size

Returns:

error-code

15.60.2.13 `int mifi_project_values(const char *proj_input, const char *proj_output, double *in_out_x_vals, double *in_out_y_vals, const int num)`

project values so that the projection $(x,y) \Rightarrow (x_{proj}, y_{proj})$ can be expressed as $x_{proj}(x,y)$, $y_{proj}(x,y)$
 all values must be given or will be returned in radians when converted from/to latlon

Parameters:

proj_input input projection proj string
proj_output output projection proj string
in_out_x_vals x-values, will be input and output
in_out_y_vals y-values, will be input and output
num size of arrays

Returns:

error-code

15.60.2.14 `int mifi_vector_reproject_values_by_matrix_f(int method, const double *matrix, float *u_out, float *v_out, int ox, int oy, int oz)`

calculate the reprojected vectors with a known matrix for [mifi_vector_reproject_values_f](#)

Parameters:

method (one of MIFI_VECTOR_KEEP_SIZE, MIFI_VECTOR_RESIZE)
matrix reprojection matrix of size (4,ox,oy)
u_out values of u, with position in the output-projection (i.e. by previously applying `mifi_interpolate_f`). The values here will be changed!
v_out values of v, with position in the output-projection (i.e. by previously applying `mifi_interpolate_f`). The values here will be changed!
ox x-dimension of outfield
oy y-dimension of outfield

oz z-dimension of the outfield

Returns:

MIFI_OK or error value

15.60.2.15 `int mifi_vector_reproject_values_f(int method, const char *proj_input, const char *proj_output, float *u_out, float *v_out, const double *out_x_axis, const double *out_y_axis, int out_x_axis_type, int out_y_axis_type, int ox, int oy, int oz)`

interpolate the vector values

When reprojecting a vector (i.e. wind (u, v)) from one projection to another, not only the base-position of the vector will change, but also the angle of the vector might change due to rotation and stretching within the projection. Thus, the values of (u,v) have to be changed accordingly to projection.

This function allows to only rotate the vector values (MIFI_VECTOR_KEEP_SIZE) which is useful to keep the windspeed constant, even if the projected plane has a different scale, or to completely reproject the vector (MIFI_VECTOR_RESIZE).

This function is implemented by using a first order tailor expansion of the projection: $(u', v') = A(u, v)$ with A a matrix defined at each point (x,y) through

```
proj(x,y)_x' = a11*x+a21*y
proj(x,y)_y' = a12*x+a22*y
```

and the same formulars for (x+delta, y) and (x, y+delta) (with delta a small value against the x or y)

Parameters:

method (one of MIFI_VECTOR_KEEP_SIZE, MIFI_VECTOR_RESIZE)

proj_input proj4-string of projection of infield

proj_output proj4-string of projection of outfield

u_out values of u, with position in the output-projection (i.e. by previously applying mifi_interpolate_f). The values here will be changed!

v_out values of v, with position in the output-projection (i.e. by previously applying mifi_interpolate_f). The values here will be changed!

out_x_axis field of size ox. Axis needs to be strong monotonous and if longitude/latitude in degree

out_y_axis field of size oy. Axis needs to be strong monotonous and if longitude/latitude in degree

out_x_axis_type one of MIFI_LATITUDE, MIFI_LONGITUDE, MIFI_PROJ_AXIS

out_y_axis_type one of MIFI_LATITUDE, MIFI_LONGITUDE, MIFI_PROJ_AXIS

ox x-dimension of outfield

oy y-dimension of outfield

oz z-dimension of the outfield

Returns:

MIFI_OK or error value

15.61 include/fimex/Logger.h File Reference

```
#include <boost/shared_ptr.hpp>
#include <string>
#include <sstream>
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::Logger](#)

Defines

- #define [LOG4FIMEX](#)(logger, level, message)

Typedefs

- typedef boost::shared_ptr< Logger > [MetNoFimex::LoggerPtr](#)

Functions

- Logger::LogLevel [MetNoFimex::defaultLogLevel](#) ()
- void [MetNoFimex::defaultLogLevel](#) (Logger::LogLevel)
- LoggerPtr [MetNoFimex::getLogger](#) (const **std::string** &className)

15.61.1 Define Documentation

15.61.1.1 #define LOG4FIMEX(logger, level, message)

Value:

```
{ \
    if (logger->isEnabledFor(level)) {\
        std::ostringstream buffer; \
        buffer << message; \
        logger->forcedLog(level, buffer.str(), __FILE__, __LINE__);}
```

use this pragma to log a message of a level

Parameters:

logger a logger as retrieved with `getLogger("com.bar")`

level a fimex LogLevel, i.e. OFF, FATAL, ERROR, WARN, INFO, DEBUG

message the message to log

15.62 include/fimex/mifi_cdm_reader.h File Reference

```
#include <boost/shared_ptr.hpp>
#include "fimex/CDMReader.h"
```

Classes

- class [mifi_cdm_reader](#)

15.63 include/fimex/NcmlCDMReader.h File Reference

```
#include "fimex/CDMReader.h"  
#include "fimex/CDMDataType.h"  
#include "boost/noncopyable.hpp"  
#include <map>
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::NcmlCDMReader](#)

15.64 include/fimex/NetCDF_CDMReader.h File Reference

```
#include "fimex/CDMReader.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::NetCDF_CDMReader](#)

15.65 include/fimex/NetCDF_CDMWriter.h File Reference

```
#include "fimex/CDMWriter.h"
#include "fimex/CDM.h"
#include <map>
#include <string>
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::NetCDF_CDMWriter](#)

15.66 include/fimex/Null_CDMWriter.h File Reference

```
#include "fimex/CDMWriter.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::Null_CDMWriter](#)

15.67 include/fimex/ReplaceStringObject.h File Reference

```
#include <iostream>
#include <string>
#include <vector>
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::ReplaceStringObject](#)

15.68 include/fimex/ReplaceStringTimeObject.h File Reference

```
#include "fimex/ReplaceStringObject.h"  
#include <ctime>
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::ReplaceStringTimeObject](#)

15.69 include/fimex/SliceBuilder.h File Reference

```
#include <string>
#include <vector>
#include <map>
#include <boost/shared_ptr.hpp>
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::SliceBuilder](#)

15.70 include/fimex/SpatialAxisSpec.h File Reference

```
#include "fimex/TimeUnit.h"  
#include "fimex/CDMException.h"  
#include <vector>  
#include <string>
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::SpatialAxisSpec](#)

15.71 `include/fimex/TimeLevelDataSliceFetcher.h` File Reference

```
#include <boost/shared_ptr.hpp>
#include <vector>
#include "fimex/Logger.h"
#include "fimex/CDMException.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::TimeLevelDataSliceFetcher](#)
read a slice of a given time/level combination from a `cdmReader`

15.72 include/fimex/TimeSpec.h File Reference

```
#include "fimex/TimeUnit.h"  
#include "fimex/CDMException.h"  
#include <vector>  
#include <string>
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::TimeSpec](#)

15.73 include/fimex/TimeUnit.h File Reference

```
#include "boost/shared_ptr.hpp"
#include "fimex/Units.h"
#include "fimex/CDMException.h"
#include <iostream>
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::FimexTime](#)
- class [MetNoFimex::TimeUnit](#)

Functions

- **std::ostream** & [MetNoFimex::operator<<](#) (**std::ostream** &out, const FimexTime &fTime)
- FimexTime [MetNoFimex::string2FimexTime](#) (const **std::string** &str) throw (CDMException)

15.74 include/fimex/Units.h File Reference

```
#include <string>
#include "fimex/CDMException.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::UnitException](#)
- class [MetNoFimex::Units](#)

Functions

- void [MetNoFimex::handleUdUnitError](#) (int unitErrCode, const **std::string** &message="") throw (UnitException)

15.75 include/fimex/Utils.h File Reference

```
#include <vector>
#include <sstream>
#include <boost/date_time/posix_time/posix_time_types.hpp>
#include "fimex/CDMException.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- struct [MetNoFimex::staticCast< OUT >](#)
- class [MetNoFimex::ScaleValue< IN, OUT >](#)
- class [MetNoFimex::ChangeMissingValue< IN, OUT >](#)

Typedefs

- typedef long [MetNoFimex::epoch_seconds](#)

Functions

- int [MetNoFimex::round](#) (double num)
- **std::string** [MetNoFimex::trim](#) (const **std::string** &str)
- template<class InputIterator>
std::string [MetNoFimex::join](#) (InputIterator start, InputIterator end, **std::string** delim=",")
- template<class InputIterator>
std::string [MetNoFimex::joinPtr](#) (InputIterator start, InputIterator end, **std::string** delim=",")
- **std::vector< std::string >** [MetNoFimex::tokenize](#) (const **std::string** &str, const **std::string** &delimiters=" ")
- **std::string** [MetNoFimex::string2lowerCase](#) (const **std::string** &str)
- template<typename T>
std::string [MetNoFimex::type2string](#) (T in)
- template<>
std::string [MetNoFimex::type2string< double >](#) (double in)
- template<typename T>
T [MetNoFimex::string2type](#) (**std::string** s)
- epoch_seconds [MetNoFimex::posixTime2epochTime](#) (const boost::posix_time::ptime &time)
- template<typename T>
std::vector< T > [MetNoFimex::tokenizeDotted](#) (const **std::string** &str, const **std::string** &delimiter=",") throw (CDMException)

15.76 include/fimex/XMLDoc.h File Reference

```
#include <boost/utility.hpp>
#include <boost/shared_ptr.hpp>
#include <string>
#include "fimex/CDMException.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::XMLDoc](#)

Typedefs

- typedef struct _xmlDoc [xmlDoc](#)
- typedef struct _xmlNode [xmlNode](#)
- typedef struct _xmlXPathContext [xmlXPathContext](#)
- typedef [xmlNode](#) * [xmlNodePtr](#)
- typedef struct _xmlXPathObject [xmlXPathObject](#)
- typedef boost::shared_ptr< [xmlXPathObject](#) > [MetNoFimex::XPathObjPtr](#)

Functions

- [std::string MetNoFimex::getXmlProp](#) (const [xmlNodePtr](#) node, const **std::string** &attrName)
- [std::string MetNoFimex::getXmlName](#) (const [xmlNodePtr](#) node)

15.76.1 Typedef Documentation

15.76.1.1 typedef struct _xmlDoc xmlDoc

15.76.1.2 typedef struct _xmlNode xmlNode

15.76.1.3 typedef xmlNode* xmlNodePtr

15.76.1.4 typedef struct _xmlXPathContext xmlXPathContext

15.76.1.5 typedef struct _xmlXPathObject xmlXPathObject

Chapter 16

Example Documentation

16.1 coordinateSystem.cpp

Example on using the CoordinateSystem in combination with a CDMReader.

```
#include "fimex/coordSys/CoordinateSystem.h"
#include "fimex/NetCDF_CDMReader.h"

int main(int argc, char* args[]) {
    boost::shared_ptr<CDMReader> reader(new NetCDF_CDMReader("coordTest.nc"));
    const CDM& cdm = reader->getCDM();

    // get all coordinate systems from file, usually one, but may be a few (theoretical limit: # of variables)
    vector<boost::shared_ptr<const CoordinateSystem> > coordSys = listCoordinateSystems(cdm);
    // find an appropriate coordinate system for a variable
    string varName = "altitude";
    vector<boost::shared_ptr<const CoordinateSystem> >::iterator varSysIt =
        find_if(coordSys.begin(), coordSys.end(), CompleteCoordinateSystemForComparator(varName));
    if (varSysIt != coordSys.end()) {
        if ((*varSysIt)->isSimpleSpatialGridded()) {
            CoordinateSystem::ConstAxisPtr xAxis = (*varSysIt)->getGeoXAxis(); // X or Lon
            CoordinateSystem::ConstAxisPtr yAxis = (*varSysIt)->getGeoYAxis(); // Y or Lat
            CoordinateSystem::ConstAxisPtr tAxis = (*varSysIt)->getTimeAxis(); // time
            // create a slice-builder for the variable
            // the slicebuilder starts with the maximum variable size
            SliceBuilder sb(cdm, altitude);
            // select 3-7 y-points
            sb.setStartAndSize(yAxis, 3, 5);
            // select the 2nd time slice
            sb.setStartAndSize(tAxis, 2, 2);

            // fetch the data
            boost::shared_ptr<Data> data = reader->getDataSlice(altitude, sb);
            /* do something with the data */
        }
    }
    return 0;
}
```

Index

- ~CDM
 - MetNoFimex::CDM, [55](#)
- ~CDMAttribute
 - MetNoFimex::CDMAttribute, [67](#)
- ~CDMDimension
 - MetNoFimex::CDMDimension, [69](#)
- ~CDMExtractor
 - MetNoFimex::CDMExtractor, [72](#)
- ~CDMInterpolator
 - MetNoFimex::CDMInterpolator, [77](#)
- ~CDMNameEqual
 - MetNoFimex::CDMNameEqual, [82](#)
- ~CDMNameEqualPtr
 - MetNoFimex::CDMNameEqualPtr, [83](#)
- ~CDMNamedEntity
 - MetNoFimex::CDMNamedEntity, [81](#)
- ~CDMQualityExtractor
 - MetNoFimex::CDMQualityExtractor, [85](#)
- ~CDMReader
 - MetNoFimex::CDMReader, [88](#)
- ~CDMTimeInterpolator
 - MetNoFimex::CDMTimeInterpolator, [92](#)
- ~CDMVariable
 - MetNoFimex::CDMVariable, [95](#)
- ~CDMWriter
 - MetNoFimex::CDMWriter, [97](#)
- ~C_CDMReader
 - MetNoFimex::C_CDMReader, [45](#)
- ~CachedForwardInterpolation
 - MetNoFimex::CachedForwardInterpolation, [47](#)
- ~CachedInterpolation
 - MetNoFimex::CachedInterpolation, [48](#)
- ~CachedVectorReprojection
 - MetNoFimex::CachedVectorReprojection, [51](#)
- ~CompleteCoordinateSystemForComparator
 - MetNoFimex::CompleteCoordinateSystemForComparator, [99](#)
- ~CoordinateAxis
 - MetNoFimex::CoordinateAxis, [101](#)
- ~CoordinateSystem
 - MetNoFimex::CoordinateSystem, [103](#)
- ~Data
 - MetNoFimex::Data, [108](#)
- ~DataImpl
 - MetNoFimex::DataImpl, [114](#)
- ~DataTypeChanger
 - MetNoFimex::DataTypeChanger, [120](#)
- ~FeltCDMReader
 - MetNoFimex::FeltCDMReader, [140](#)
- ~FeltCDMReader2
 - MetNoFimex::FeltCDMReader2, [141](#)
- ~FeltField
 - felt::FeltField, [143](#)
- ~FeltFile
 - felt::FeltFile, [146](#)
- ~FeltGridDefinition
 - felt::FeltGridDefinition, [148](#)
- ~FeltParameters
 - MetNoFelt::FeltParameters, [150](#)
- ~Felt_Array
 - MetNoFelt::Felt_Array, [123](#)
- ~Felt_Array2
 - MetNoFelt::Felt_Array2, [128](#)
- ~Felt_File
 - MetNoFelt::Felt_File, [132](#)
- ~Felt_File2
 - MetNoFelt::Felt_File2, [136](#)
- ~GribApiCDMWriter
 - MetNoFimex::GribApiCDMWriter, [155](#)
- ~GribApiCDMWriter_Impl1
 - MetNoFimex::GribApiCDMWriter_Impl1, [156](#)
- ~GribApiCDMWriter_Impl2
 - MetNoFimex::GribApiCDMWriter_Impl2, [158](#)
- ~GribApiCDMWriter_ImplAbstract
 - MetNoFimex::GribApiCDMWriter_ImplAbstract, [161](#)
- ~GribCDMReader
 - MetNoFimex::GribCDMReader, [164](#)
- ~GribFileIndex
 - MetNoFimex::GribFileIndex, [165](#)
- ~GribFileMessage
 - MetNoFimex::GribFileMessage, [166](#)
- ~GribFileMessageEqualLevelTime
 - MetNoFimex::GribFileMessageEqualLevelTime, [169](#)
- ~GribFileMessageEqualTime
 - MetNoFimex::GribFileMessageEqualTime, [169](#)

- MetNoFimex::GribFileMessageEqualTime, 170
- ~GridDefinition
 - MetNoFimex::GridDefinition, 173
- ~LambertConformalConicProjection
 - MetNoFimex::LambertConformalConicProjection, 176
- ~LatitudeLongitudeProjection
 - MetNoFimex::LatitudeLongitudeProjection, 178
- ~Logger
 - MetNoFimex::Logger, 180
- ~MercatorProjection
 - MetNoFimex::MercatorProjection, 182
- ~NcmlCDMReader
 - MetNoFimex::NcmlCDMReader, 186
- ~NetCDF_CDMReader
 - MetNoFimex::NetCDF_CDMReader, 187
- ~NetCDF_CDMWriter
 - MetNoFimex::NetCDF_CDMWriter, 189
- ~Null_CDMWriter
 - MetNoFimex::Null_CDMWriter, 191
- ~PolarStereographicProjection
 - MetNoFimex::PolarStereographicProjection, 192
- ~Projection
 - MetNoFimex::Projection, 195
- ~ProjectionImpl
 - MetNoFimex::ProjectionImpl, 198
- ~ReplaceStringObject
 - MetNoFimex::ReplaceStringObject, 201
- ~ReplaceStringTimeObject
 - MetNoFimex::ReplaceStringTimeObject, 203
- ~RotatedLatitudeLongitudeProjection
 - MetNoFimex::RotatedLatitudeLongitudeProjection, 206
- ~SliceBuilder
 - MetNoFimex::SliceBuilder, 209
- ~SpatialAxisSpec
 - MetNoFimex::SpatialAxisSpec, 212
- ~StereographicProjection
 - MetNoFimex::StereographicProjection, 214
- ~TimeLevelDataSliceFetcher
 - MetNoFimex::TimeLevelDataSliceFetcher, 216
- ~TimeSpec
 - MetNoFimex::TimeSpec, 218
- ~TimeUnit
 - MetNoFimex::TimeUnit, 220
- ~Units
 - MetNoFimex::Units, 222
- ~XMLDoc
 - MetNoFimex::XMLDoc, 224
- acceptsProj4
 - MetNoFimex::LambertConformalConicProjection, 176
 - MetNoFimex::LatitudeLongitudeProjection, 178
 - MetNoFimex::MercatorProjection, 182
 - MetNoFimex::PolarStereographicProjection, 192
 - MetNoFimex::RotatedLatitudeLongitudeProjection, 206
 - MetNoFimex::StereographicProjection, 214
- addAttribute
 - MetNoFimex::CDM, 59
- addDimension
 - MetNoFimex::CDM, 57
- addIdent19
 - MetNoFelt::Felt_Array, 125
- addInformationByField
 - MetNoFelt::Felt_Array2, 128
- addInformationByIndex
 - MetNoFelt::Felt_Array, 123
- addOrReplaceAttribute
 - MetNoFimex::CDM, 59
- addParameter
 - MetNoFimex::Projection, 195
 - MetNoFimex::ProjectionImpl, 198
- addParameters
 - MetNoFimex::Projection, 195
 - MetNoFimex::ProjectionImpl, 198
- addParameterToStream
 - MetNoFimex::ProjectionImpl, 199
- addVariable
 - MetNoFimex::CDM, 55
- ANY_ARRAY
 - MetNoFelt, 31
- ANY_ARRAY20
 - MetNoFelt, 31
- ANY_VALUE
 - MetNoFelt, 31
- areConvertible
 - MetNoFimex::Units, 222
- as
 - MetNoFimex::DataImpl, 114, 115
- asBase
 - MetNoFimex::DataImpl, 114
- asChar
 - MetNoFimex::Data, 109
 - MetNoFimex::DataImpl, 115
- asConstChar
 - MetNoFimex::Data, 109
 - MetNoFimex::DataImpl, 115
- asConstDouble
 - MetNoFimex::Data, 110
 - MetNoFimex::DataImpl, 116

- asConstFloat
 - MetNoFimex::Data, [109](#)
 - MetNoFimex::DataImpl, [115](#)
- asConstInt
 - MetNoFimex::Data, [109](#)
 - MetNoFimex::DataImpl, [115](#)
- asConstShort
 - MetNoFimex::Data, [109](#)
 - MetNoFimex::DataImpl, [115](#)
- asDouble
 - MetNoFimex::Data, [110](#)
 - MetNoFimex::DataImpl, [116](#)
- asFloat
 - MetNoFimex::Data, [110](#)
 - MetNoFimex::DataImpl, [116](#)
- asInt
 - MetNoFimex::Data, [109](#)
 - MetNoFimex::DataImpl, [115](#)
- asShort
 - MetNoFimex::Data, [109](#)
 - MetNoFimex::DataImpl, [115](#)
- asString
 - MetNoFimex::Data, [110](#)
 - MetNoFimex::DataImpl, [116](#)
- at
 - felt::FeltFile, [146](#)
- AttrVec
 - MetNoFimex::CDM, [55](#)
- AxisPtr
 - MetNoFimex::CoordinateSystem, [102](#)
- AxisType
 - MetNoFimex::CoordinateAxis, [100](#)
- begin
 - felt::FeltFile, [146](#)
- binary, [43](#)
 - value, [43](#)
- binary < 0 >, [44](#)
 - value, [44](#)
- blockSize
 - felt, [29](#)
- blockWords
 - felt, [29](#)
- BOOST_STATIC_ASSERT
 - FeltConstants.h, [228](#)
- bytes_for_one
 - MetNoFimex::Data, [108](#)
 - MetNoFimex::DataImpl, [114](#)
- C_CDMReader
 - MetNoFimex::C_CDMReader, [45](#)
- c_fimex.h
 - doubleDatasliceCallbackPtr, [236](#)
 - mifi_cdm_reader, [236](#)
 - mifi_free_cdm_reader, [237](#)
 - mifi_get_double_data, [237](#)
 - mifi_get_double_datasize, [237](#)
 - mifi_get_variable_name, [237](#)
 - mifi_get_variable_number, [238](#)
 - mifi_grib_writer, [238](#)
 - mifi_netcdf_writer, [238](#)
 - mifi_new_c_reader, [238](#)
 - mifi_new_cdminterpolator, [239](#)
 - mifi_new_felt_reader, [239](#)
 - mifi_new_grib_reader, [239](#)
 - mifi_new_ncml_modifier, [240](#)
 - mifi_new_ncml_reader, [240](#)
 - mifi_new_netcdf_reader, [240](#)
 - mifi_nullcdm_writer, [240](#)
 - mifi_set_callback_double, [240](#)
- CachedForwardInterpolation
 - MetNoFimex::CachedForwardInterpolation, [47](#)
- CachedInterpolation
 - MetNoFimex::CachedInterpolation, [48](#)
- CachedVectorReprojection
 - MetNoFimex::CachedVectorReprojection, [51](#)
- CDM
 - MetNoFimex::CDM, [55](#)
- CDM_CHAR
 - MetNoFimex, [35](#)
- CDM_DOUBLE
 - MetNoFimex, [36](#)
- CDM_FLOAT
 - MetNoFimex, [36](#)
- CDM_INT
 - MetNoFimex, [36](#)
- CDM_NAT
 - MetNoFimex, [35](#)
- CDM_SHORT
 - MetNoFimex, [36](#)
- CDM_STRING
 - MetNoFimex, [36](#)
- cdm_
 - MetNoFimex::CDMReader, [91](#)
- CDMAttribute
 - MetNoFimex::CDMAttribute, [66](#), [67](#)
- CDMconstants.h
 - fimexHasFelt, [247](#)
 - fimexHasGribApi, [247](#)
 - fimexHasNetcdf, [247](#)
 - fimexVersion, [248](#)
 - MIFI_EARTH_RADIUS_M, [247](#)
 - MIFI_FILETYPE_FELT, [247](#)
 - MIFI_FILETYPE_GRIB, [247](#)
 - MIFI_FILETYPE_NCML, [247](#)
 - MIFI_FILETYPE_NETCDF, [247](#)
 - MIFI_FILETYPE_UNKNOWN, [247](#)

- CDMDataType
 - MetNoFimex, 35
- CDMDimension
 - MetNoFimex::CDMDimension, 69
- CDMException
 - MetNoFimex::CDMException, 71
- CDMExtractor
 - MetNoFimex::CDMExtractor, 72
- CDMInterpolator
 - MetNoFimex::CDMInterpolator, 77
- CDMNameEqual
 - MetNoFimex::CDMNameEqual, 82
- CDMNameEqualPtr
 - MetNoFimex::CDMNameEqualPtr, 83
- CDMQualityExtractor
 - MetNoFimex::CDMQualityExtractor, 84
- CDMReader
 - MetNoFimex::CDMReader, 88
- cdmReader
 - MetNoFimex::CDMWriter, 97
- CDMTimeInterpolator
 - MetNoFimex::CDMTimeInterpolator, 92
- CDMVariable
 - MetNoFimex::CDMVariable, 95
- CDMWriter
 - MetNoFimex::CDMWriter, 97
- changeDataType
 - MetNoFimex::CDMExtractor, 74
- ChangeMissingValue
 - MetNoFimex::ChangeMissingValue, 98
- changeProjection
 - MetNoFimex::CDMInterpolator, 77, 78
- changeTimeAxis
 - MetNoFimex::CDMTimeInterpolator, 92
- checkDimension
 - MetNoFimex::CDMVariable, 95
- checkVariableAttribute
 - MetNoFimex::CDM, 57
- comparableTo
 - MetNoFimex::GridDefinition, 174
- CompleteCoordinateSystemForComparator
 - MetNoFimex::CompleteCoordinateSystemForComparator, 99
- configFile
 - MetNoFimex::GribApiCDMWriter_ - ImplAbstract, 163
- const_iterator
 - felt::FeltFile, 146
- ConstAxisList
 - MetNoFimex::CoordinateSystem, 103
- ConstAxisPtr
 - MetNoFimex::CoordinateSystem, 102
- constConvertArrayType
 - MetNoFimex, 36
- contentSummary
 - felt, 28
- convert
 - MetNoFimex::Units, 222
- convertArrayType
 - MetNoFimex, 36
- convertData
 - MetNoFimex::DataTypeChanger, 121
- convertDataType
 - MetNoFimex::Data, 111
 - MetNoFimex::DataImpl, 117
- CoordinateAxis
 - MetNoFimex::CoordinateAxis, 101
- CoordinateSystem
 - MetNoFimex::CoordinateSystem, 103
- create
 - MetNoFimex::CDMFileReaderFactory, 75
 - MetNoFimex::Projection, 196
- createByProj4
 - MetNoFimex::Projection, 196
- createData
 - MetNoFimex, 36
- createDataSlice
 - MetNoFimex, 37
- DataImpl
 - MetNoFimex::DataImpl, 114
- dataType
 - felt::FeltField, 143
- datatype2string
 - MetNoFimex, 37
- DataTypeChanger
 - MetNoFimex::DataTypeChanger, 120
- dataVersion
 - felt::FeltField, 144
- DEBUG
 - MetNoFimex::Logger, 180
- DEFAULT_CONFIG
 - MetNoFelt::FeltParameters, 151
- defaultLogLevel
 - MetNoFimex, 37
- DEPRECATED
 - deprecated.h, 275
 - MetNoFimex, 37
 - MetNoFimex::CDM, 62
- deprecated.h
 - DEPRECATED, 275
- detectFileType
 - MetNoFimex::CDMFileReaderFactory, 75
- DimVec
 - MetNoFimex::CDM, 55
- doubleDatasliceCallbackPtr
 - c_fimex.h, 236
- doxydoc.txt, 227

- duplicateArrayType
 - MetNoFimex, 38
- EARTH_RADIUS
 - felt, 29
- empty
 - felt::FeltFile, 146
- end
 - felt::FeltFile, 146
- epoch_seconds
 - MetNoFimex, 35
- epochSeconds2unitTime
 - MetNoFimex::TimeUnit, 220
- ERROR
 - MetNoFimex::Logger, 180
- FATAL
 - MetNoFimex::Logger, 180
- felt, 27
 - blockSize, 29
 - blockWords, 29
 - contentSummary, 28
 - EARTH_RADIUS, 29
 - FeltGridDefinitionPtr, 28
 - get, 28
 - gridParameters, 28
 - gridParametersToProjDefinition, 28
 - isUndefined, 28
 - offsetToContentDefinition, 29
 - parseTime, 28
 - parseTimeNoThrow, 29
 - PI, 29
 - word, 28
- felt::FeltField, 142
 - ~FeltField, 143
 - dataType, 143
 - dataVersion, 144
 - FeltField, 143
 - getHeader, 144
 - grid, 143
 - gridArea, 143
 - gridInformation, 144
 - gridSize, 143
 - gridType, 143
 - Header, 143
 - information, 144
 - level1, 143
 - level2, 143
 - miscField, 144
 - parameter, 143
 - producer, 143
 - projectionInformation, 144
 - referenceTime, 143
 - scaleFactor, 143
 - valid, 143
 - validTime, 143
 - verticalCoordinate, 143
 - xNum, 144
 - yNum, 144
- felt::FeltFile, 145
 - ~FeltFile, 146
 - at, 146
 - begin, 146
 - const_iterator, 146
 - empty, 146
 - end, 146
 - FeltField, 147
 - FeltFieldPtr, 146
 - FeltFile, 146
 - fileName, 146
 - firstTime, 146
 - information, 146
 - isLogging, 147
 - iterator, 146
 - lastTime, 146
 - lastUpdateTime, 146
 - log, 146
 - referenceTime, 146
 - setLogging, 147
 - setLogStream, 147
 - size, 146
 - size_type, 146
- felt::FeltGridDefinition, 148
 - ~FeltGridDefinition, 148
 - FeltGridDefinition, 148
 - getGridParameters, 149
 - getScanMode, 149
 - getXIncrement, 149
 - getXNumber, 149
 - getYIncrement, 149
 - getYNumber, 149
 - LeftLowerHorizontal, 148
 - LeftUpperHorizontal, 148
 - Orientation, 148
 - projDefinition, 149
 - startLatitude, 149
 - startLongitude, 149
 - startX, 149
 - startY, 149
- Felt_Array
 - MetNoFelt::Felt_Array, 123
- Felt_Array2
 - MetNoFelt::Felt_Array2, 127
- Felt_File
 - MetNoFelt::Felt_File, 132
- Felt_File2
 - MetNoFelt::Felt_File2, 136
- Felt_File_Error

- MetNoFelt::Felt_File_Error, 139
- FeltCDMReader
 - MetNoFimex::FeltCDMReader, 140
- FeltCDMReader2
 - MetNoFimex::FeltCDMReader2, 141
- FeltConstants.h
 - BOOST_STATIC_ASSERT, 228
 - M_PI, 228
- FeltField
 - felt::FeltField, 143
 - felt::FeltFile, 147
- FeltFieldPtr
 - felt::FeltFile, 146
- FeltFile
 - felt::FeltFile, 146
- FeltGridDefinition
 - felt::FeltGridDefinition, 148
- FeltGridDefinitionPtr
 - felt, 28
- FeltParameters
 - MetNoFelt::FeltParameters, 150
- fileName
 - felt::FeltFile, 146
- fimexHasFelt
 - CDMconstants.h, 247
- fimexHasGribApi
 - CDMconstants.h, 247
- fimexHasNetcdf
 - CDMconstants.h, 247
- fimexTime2unitTime
 - MetNoFimex::TimeUnit, 220
- fimexTime2unitTimeX
 - MetNoFimex::TimeUnit, 220
- fimexVersion
 - CDMconstants.h, 248
- findAxisOfType
 - MetNoFimex::CoordinateSystem, 105
- findVariables
 - MetNoFimex::CDM, 56
- firstTime
 - felt::FeltFile, 146
- forcedLog
 - MetNoFimex::Logger, 180
- generateProjectionCoordinates
 - MetNoFimex::CDM, 62
- GeoX
 - MetNoFimex::CoordinateAxis, 100
- GeoY
 - MetNoFimex::CoordinateAxis, 100
- GeoZ
 - MetNoFimex::CoordinateAxis, 100
- get
 - felt, 28
 - mifi_cdm_reader, 184
- getAttribute
 - MetNoFimex::CDM, 60, 61
 - MetNoFimex::NetCDF_CDMWriter, 190
- getAttributes
 - MetNoFimex::CDM, 60
- getAxes
 - MetNoFimex::CoordinateSystem, 106
- getAxisSteps
 - MetNoFimex::SpatialAxisSpec, 212
- getAxisType
 - MetNoFimex::CoordinateAxis, 101
- getAxisTypeStr
 - MetNoFimex::CoordinateAxis, 101
- getCDM
 - MetNoFimex::CDMReader, 88
- getConventionName
 - MetNoFimex::CoordinateSystem, 103
- getData
 - MetNoFimex::CDMAttribute, 67
 - MetNoFimex::CDMReader, 88
 - MetNoFimex::CDMVariable, 96
- getDataHeader
 - MetNoFelt::Felt_Array, 124
- getDataPtr
 - MetNoFimex::Data, 108
 - MetNoFimex::DataImpl, 114
- getDataSlice
 - MetNoFelt::Felt_File, 132
 - MetNoFimex::C_CDMReader, 45
 - MetNoFimex::CDMExtractor, 72
 - MetNoFimex::CDMInterpolator, 77
 - MetNoFimex::CDMQualityExtractor, 85
 - MetNoFimex::CDMReader, 88
 - MetNoFimex::CDMTimeInterpolator, 92
 - MetNoFimex::FeltCDMReader, 140
 - MetNoFimex::FeltCDMReader2, 141
 - MetNoFimex::GribCDMReader, 164
 - MetNoFimex::NcmlCDMReader, 186
 - MetNoFimex::NetCDF_CDMReader, 187, 188
- getDataSliceFromMemory
 - MetNoFimex::CDMReader, 90
- getDataType
 - MetNoFimex::CDMAttribute, 67
 - MetNoFimex::CDMVariable, 95
 - MetNoFimex::Data, 111
 - MetNoFimex::DataImpl, 118
 - MetNoFimex::DataTypeChanger, 121
- getDatatype
 - MetNoFelt::Felt_Array, 124
 - MetNoFelt::Felt_Array2, 128
- getDateTime
 - MetNoFimex::GribFileMessage, 167

- getDimension
 - MetNoFimex::CDM, 58
- getDimensionName
 - MetNoFimex::NetCDF_CDMWriter, 189
- getDimensionNames
 - MetNoFimex::SliceBuilder, 210
- getDimensions
 - MetNoFimex::CDM, 60
- getDimensionSizes
 - MetNoFimex::SliceBuilder, 210
- getDimensionStartPositions
 - MetNoFimex::SliceBuilder, 210
- getEdition
 - MetNoFimex::GribFileMessage, 167
- getFeltArray
 - MetNoFelt::Felt_File, 132
 - MetNoFelt::Felt_File2, 136
- getFeltLevelPairs
 - MetNoFelt::Felt_File, 133
 - MetNoFelt::Felt_File2, 137
- getFeltLevels
 - MetNoFelt::Felt_File, 133
- getFeltReader
 - MetNoFimex, 38
- getFeltTimes
 - MetNoFelt::Felt_File, 133
 - MetNoFelt::Felt_File2, 137
- getField
 - MetNoFelt::Felt_Array2, 130
- getFieldSize
 - MetNoFelt::Felt_Array, 126
- getFilePosition
 - MetNoFimex::GribFileMessage, 167
- getFileURL
 - MetNoFimex::GribFileMessage, 167
- getFillValue
 - MetNoFelt::Felt_Array, 124
 - MetNoFelt::Felt_Array2, 129
 - MetNoFimex::CDM, 61
- getGeoXAxis
 - MetNoFimex::CoordinateSystem, 105
- getGeoYAxis
 - MetNoFimex::CoordinateSystem, 105
- getGeoZAxis
 - MetNoFimex::CoordinateSystem, 105
- getGrid
 - MetNoFelt::Felt_Array2, 128
- getGridAllowDelta
 - MetNoFelt::Felt_Array2, 128
- getGridDefinition
 - MetNoFelt::Felt_Array2, 129
 - MetNoFelt::Felt_File2, 138
 - MetNoFimex::GribFileMessage, 167
- getGridParameters
 - felt::FeltGridDefinition, 149
 - MetNoFelt::Felt_Array, 124
 - MetNoFelt::Felt_File, 134
- getGridType
 - MetNoFelt::Felt_Array, 124
 - MetNoFelt::Felt_Array2, 130
 - MetNoFelt::Felt_File, 134
 - MetNoFelt::Felt_File2, 137
- getHeader
 - felt::FeltField, 144
- getHorizontalXAxis
 - MetNoFimex::CDM, 63
- getHorizontalYAxis
 - MetNoFimex::CDM, 63
- getHybridLevels
 - MetNoFelt::Felt_File, 133
 - MetNoFelt::Felt_File2, 137
- getIdent19
 - MetNoFelt::Felt_Array, 125
 - MetNoFelt::Felt_Array2, 129
- getIndex
 - MetNoFelt::Felt_Array, 126
- getIndexHeader
 - MetNoFelt::Felt_Array, 123
- getLatitudeLongitude
 - MetNoFimex::CDM, 63
- getLatitudeName
 - MetNoFimex::CDMInterpolator, 78
- getLength
 - MetNoFimex::CDMDimension, 70
- getLevelNumber
 - MetNoFimex::GribFileMessage, 167
- getLevelPairs
 - MetNoFelt::Felt_Array, 125
 - MetNoFelt::Felt_Array2, 129
- getLevels
 - MetNoFelt::Felt_Array, 125
 - MetNoFimex::GribApiCDMWriter - ImplAbstract, 162
- getLevelType
 - MetNoFelt::Felt_Array, 124
 - MetNoFelt::Felt_Array2, 128
 - MetNoFimex::GribFileMessage, 167
- getLogger
 - MetNoFimex, 38
- getLongitudeName
 - MetNoFimex::CDMInterpolator, 79
- getMaxDimensionSizes
 - MetNoFimex::SliceBuilder, 210
- getMessageNumber
 - MetNoFimex::GribFileMessage, 167
- getName
 - MetNoFelt::Felt_Array, 124
 - MetNoFelt::Felt_Array2, 128

- MetNoFimex::CDMAttribute, 67
- MetNoFimex::CDMDimension, 69
- MetNoFimex::CDMNamedEntity, 81
- MetNoFimex::CDMVariable, 95
- MetNoFimex::GribFileMessage, 167
- MetNoFimex::Projection, 195
- MetNoFimex::ProjectionImpl, 198
- getNodePtr
 - MetNoFimex::GribApiCDMWriter_ - ImplAbstract, 162
- getNX
 - MetNoFelt::Felt_File, 133
 - MetNoFelt::Felt_File2, 137
- getNY
 - MetNoFelt::Felt_File, 134
 - MetNoFelt::Felt_File2, 137
- getParameterDatatype
 - MetNoFelt::FeltParameters, 151
- getParameterFillValue
 - MetNoFelt::FeltParameters, 151
- getParameterIds
 - MetNoFimex::GribFileMessage, 167
- getParameterName
 - MetNoFelt::FeltParameters, 151
- getParameters
 - MetNoFelt::FeltParameters, 151
 - MetNoFimex::Projection, 195
 - MetNoFimex::ProjectionImpl, 198
- getProj4ProjectionPart
 - MetNoFimex::LambertConformalConicProjection, 176
 - MetNoFimex::LatitudeLongitudeProjection, 178
 - MetNoFimex::MercatorProjection, 182
 - MetNoFimex::ProjectionImpl, 199
 - MetNoFimex::RotatedLatitudeLongitudeProjection, 206
 - MetNoFimex::StereographicProjection, 215
- getProj4String
 - MetNoFimex::Projection, 195
 - MetNoFimex::ProjectionImpl, 198
- getProjDefinition
 - MetNoFimex::GridDefinition, 173
- getProjection
 - MetNoFimex::CoordinateSystem, 104
- getProjectionOf
 - MetNoFimex::CDM, 63
- getProjString
 - MetNoFelt, 31
- getScaledData
 - MetNoFimex::CDMReader, 90
- getScaledDataInUnit
 - MetNoFimex::CDMReader, 90
- getScaledDataSlice
 - MetNoFelt::Felt_File, 133
 - MetNoFelt::Felt_File2, 136
 - MetNoFimex::CDMReader, 89
- getScaledDataSliceInUnit
 - MetNoFimex::CDMReader, 89, 90
- getScalingFactor
 - MetNoFelt::Felt_Array, 126
 - MetNoFelt::Felt_Array2, 129
- getScanMode
 - felt::FeltGridDefinition, 149
 - MetNoFimex::GridDefinition, 174
- getShape
 - MetNoFimex::CDMVariable, 95
- getShortName
 - MetNoFimex::GribFileMessage, 167
- getSpatialVectorCounterpart
 - MetNoFimex::CDMVariable, 95
- getSpatialVectorDirection
 - MetNoFimex::CDMVariable, 95
- getStatusVariable
 - MetNoFimex::CDMQualityExtractor, 85
- getStringValue
 - MetNoFimex::CDMAttribute, 67
- getTimeAxis
 - MetNoFimex::CDM, 64
 - MetNoFimex::CoordinateSystem, 106
- getTimeLevelSlice
 - MetNoFimex::TimeLevelDataSliceFetcher, 216
- getTimeSteps
 - MetNoFelt::Felt_Array, 125
 - MetNoFelt::Felt_Array2, 129
 - MetNoFimex::GribApiCDMWriter_ - ImplAbstract, 162
- getTimeSpec
 - MetNoFimex::TimeSpec, 218
- getTypeOfGrid
 - MetNoFimex::GribFileMessage, 167
- getUnits
 - MetNoFimex::CDM, 61
- getUnitString
 - MetNoFimex::TimeSpec, 218
- getUnlimitedDim
 - MetNoFimex::CDM, 59
- getUrl
 - MetNoFimex::GribFileIndex, 165
- getVariable
 - MetNoFimex::CDM, 55
- getVariableFlags
 - MetNoFimex::CDMQualityExtractor, 85
- getVariableName
 - MetNoFimex::NetCDF_CDMWriter, 189
- getVariables
 - MetNoFimex::CDM, 60

- getVariableValues
 - MetNoFimex::CDMQualityExtractor, 85
- getVerticalAxis
 - MetNoFimex::CDM, 64
- getVerticalFeltType
 - MetNoFelt::Felt_Array, 126
- getX
 - MetNoFelt::Felt_Array, 126
 - MetNoFelt::Felt_Array2, 129
- getXData
 - MetNoFelt::Felt_File, 134
 - MetNoFelt::Felt_File2, 137
- getXIncrement
 - felt::FeltGridDefinition, 149
 - MetNoFimex::GridDefinition, 173
- getXmlName
 - MetNoFimex, 38
- getXmlProp
 - MetNoFimex, 38
- getXNumber
 - felt::FeltGridDefinition, 149
- getXPathObject
 - MetNoFimex::XMLDoc, 224
- getXSize
 - MetNoFimex::CachedVectorReprojection, 51
 - MetNoFimex::GridDefinition, 173
- getXStart
 - MetNoFimex::GridDefinition, 173
- getY
 - MetNoFelt::Felt_Array, 126
 - MetNoFelt::Felt_Array2, 129
- getYData
 - MetNoFelt::Felt_File, 134
 - MetNoFelt::Felt_File2, 137
- getYIncrement
 - felt::FeltGridDefinition, 149
 - MetNoFimex::GridDefinition, 173
- getYNumber
 - felt::FeltGridDefinition, 149
- getYSize
 - MetNoFimex::CachedVectorReprojection, 51
 - MetNoFimex::GridDefinition, 173
- getYStart
 - MetNoFimex::GridDefinition, 174
- globalAttributeNS
 - MetNoFimex::CDM, 60
- GribApiCDMWriter
 - MetNoFimex::GribApiCDMWriter, 155
- GribApiCDMWriter_Impl1
 - MetNoFimex::GribApiCDMWriter_Impl1, 156
- GribApiCDMWriter_Impl2
 - MetNoFimex::GribApiCDMWriter_Impl2, 158
- GribApiCDMWriter_ImplAbstract
 - MetNoFimex::GribApiCDMWriter_ImplAbstract, 161
- GribCDMReader
 - MetNoFimex::GribCDMReader, 164
- gribDataRead
 - MetNoFimex, 38
- GribFileIndex
 - MetNoFimex::GribFileIndex, 165
- GribFileMessage
 - MetNoFimex::GribFileMessage, 166
- GribFileMessageEqualLevelTime
 - MetNoFimex::GribFileMessageEqualLevelTime, 169
- GribFileMessageEqualTime
 - MetNoFimex::GribFileMessageEqualTime, 170
- gribGetGridOrientation
 - MetNoFimex, 39
- gribHandle
 - MetNoFimex::GribApiCDMWriter_ImplAbstract, 163
- GribUtils.h
 - MIFI_GRIB_CHECK, 291
 - mifi_grib_check, 291
- gribVersion
 - MetNoFimex::GribApiCDMWriter_ImplAbstract, 163
- grid
 - felt::FeltField, 143
- gridArea
 - felt::FeltField, 143
- GridDefinition
 - MetNoFimex::GridDefinition, 173
- gridInformation
 - felt::FeltField, 144
- gridParameters
 - felt, 28
- gridParametersToProjDefinition
 - felt, 28
- gridSize
 - felt::FeltField, 143
- gridType
 - felt::FeltField, 143
- handleTypeScaleAndMissingData
 - MetNoFimex::GribApiCDMWriter_Impl1, 157
 - MetNoFimex::GribApiCDMWriter_Impl2, 159
 - MetNoFimex::GribApiCDMWriter_ImplAbstract, 162
- handleUdUnitError
 - MetNoFimex, 39

- hasAxisType
 - MetNoFimex::CoordinateSystem, 104
- hasData
 - MetNoFimex::CDMVariable, 96
- hasDimension
 - MetNoFimex::CDM, 57
- hasProjection
 - MetNoFimex::CoordinateSystem, 104
- hasUnlimitedDim
 - MetNoFimex::CDM, 59
- hasVariable
 - MetNoFimex::CDM, 56
- Header
 - felt::FeltField, 143
- Height
 - MetNoFimex::CoordinateAxis, 101
- hour
 - MetNoFimex::FimexTime, 154
- id
 - MetNoFimex::CoordinateSystem, 103
- include/felt/FeltConstants.h, 228
- include/felt/FeltField.h, 229
- include/felt/FeltFile.h, 230
- include/felt/FeltGridDefinition.h, 231
- include/felt/FeltTypeConversion.h, 232
- include/felt/FeltTypes.h, 233
- include/fimex/binaryConstants.h, 234
- include/fimex/C_CDMReader.h, 235
- include/fimex/c_fimex.h, 236
- include/fimex/CachedForwardInterpolation.h, 242
- include/fimex/CachedInterpolation.h, 243
- include/fimex/CachedVectorReprojection.h, 244
- include/fimex/CDM.h, 245
- include/fimex/CDMAttribute.h, 246
- include/fimex/CDMconstants.h, 247
- include/fimex/CDMDataType.h, 249
- include/fimex/CDMDimension.h, 250
- include/fimex/CDMException.h, 251
- include/fimex/CDMExtractor.h, 252
- include/fimex/CDMFileReaderFactory.h, 253
- include/fimex/CDMInterpolator.h, 254
- include/fimex/CDMNamedEntity.h, 255
- include/fimex/CDMQualityExtractor.h, 256
- include/fimex/CDMReader.h, 257
- include/fimex/CDMTimeInterpolator.h, 258
- include/fimex/CDMVariable.h, 259
- include/fimex/CDMWriter.h, 260
- include/fimex/coordSys/CoordinateAxis.h, 261
- include/fimex/coordSys/CoordinateSystem.h, 262
- include/fimex/coordSys/LambertConformalConicProjection.h, 263
- include/fimex/coordSys/LatitudeLongitudeProjection.h, 264
- include/fimex/coordSys/MercatorProjection.h, 265
- include/fimex/coordSys/PolarStereographicProjection.h, 266
- include/fimex/coordSys/Projection.h, 267
- include/fimex/coordSys/ProjectionImpl.h, 268
- include/fimex/coordSys/RotatedLatitudeLongitudeProjection.h, 269
- include/fimex/coordSys/StereographicProjection.h, 270
- include/fimex/Data.h, 271
- include/fimex/DataImpl.h, 272
- include/fimex/DataTypeChanger.h, 274
- include/fimex/deprecated.h, 275
- include/fimex/Felt_Array.h, 276
- include/fimex/Felt_Array2.h, 277
- include/fimex/Felt_File.h, 278
- include/fimex/Felt_File2.h, 279
- include/fimex/Felt_File_Error.h, 280
- include/fimex/Felt_Types.h, 281
- include/fimex/FeltCDMReader.h, 282
- include/fimex/FeltCDMReader2.h, 283
- include/fimex/FeltParameters.h, 284
- include/fimex/GribApiCDMWriter.h, 285
- include/fimex/GribApiCDMWriter_Impl1.h, 286
- include/fimex/GribApiCDMWriter_Impl2.h, 287
- include/fimex/GribApiCDMWriter_ImplAbstract.h, 288
- include/fimex/GribCDMReader.h, 289
- include/fimex/GribFileIndex.h, 290
- include/fimex/GribUtils.h, 291
- include/fimex/GridDefinition.h, 292
- include/fimex/interpolation.h, 293
- include/fimex/Logger.h, 304
- include/fimex/mifi_cdm_reader.h, 305
- include/fimex/NcmlCDMReader.h, 306
- include/fimex/NetCDF_CDMReader.h, 307
- include/fimex/NetCDF_CDMWriter.h, 308
- include/fimex/Null_CDMWriter.h, 309
- include/fimex/ReplaceStringObject.h, 310
- include/fimex/ReplaceStringTimeObject.h, 311
- include/fimex/SliceBuilder.h, 312
- include/fimex/SpatialAxisSpec.h, 313
- include/fimex/TimeLevelDataSliceFetcher.h, 314
- include/fimex/TimeSpec.h, 315
- include/fimex/TimeUnit.h, 316
- include/fimex/Units.h, 317
- include/fimex/Utils.h, 318
- include/fimex/XMLDoc.h, 319
- index16toLevelPair
 - MetNoFelt, 31
- index16toTime
 - MetNoFelt, 31
- INFO
 - MetNoFimex::Logger, 180

- information
 - felt::FeltField, 144
 - felt::FeltFile, 146
- interpolateValues
 - MetNoFimex::CachedForwardInterpolation, 47
 - MetNoFimex::CachedInterpolation, 49
 - MetNoFimex::CachedInterpolationInterface, 50
- interpolation.h
 - mifi_3d_array_position, 297
 - mifi_bad2nanf, 298
 - MIFI_DEBUG, 295
 - MIFI_ERROR, 295
 - mifi_get_values_bicubic_f, 298
 - mifi_get_values_bilinear_f, 298
 - mifi_get_values_f, 299
 - mifi_get_values_linear_f, 299
 - mifi_get_vector_reproject_matrix, 299
 - MIFI_INTERPOL_BICUBIC, 295
 - MIFI_INTERPOL_BILINEAR, 295
 - MIFI_INTERPOL_COORD_NN, 295
 - MIFI_INTERPOL_COORD_NN_KD, 296
 - MIFI_INTERPOL_FORWARD_MAX, 296
 - MIFI_INTERPOL_FORWARD_MEAN, 296
 - MIFI_INTERPOL_FORWARD_MEDIAN, 296
 - MIFI_INTERPOL_FORWARD_MIN, 296
 - MIFI_INTERPOL_FORWARD_SUM, 296
 - MIFI_INTERPOL_NEAREST_NEIGHBOR, 296
 - mifi_interpolate_d, 300
 - mifi_interpolate_f, 300
 - MIFI_LATITUDE, 297
 - MIFI_LONGITUDE, 297
 - mifi_nanf2bad, 301
 - MIFI_OK, 297
 - mifi_points2position, 301
 - MIFI_PROJ_AXIS, 297
 - mifi_project_axes, 301
 - mifi_project_values, 302
 - MIFI_UNDEFINED_D, 297
 - MIFI_UNDEFINED_F, 297
 - MIFI_VECTOR_KEEP_SIZE, 297
 - mifi_vector_reproject_values_by_matrix_f, 302
 - mifi_vector_reproject_values_f, 303
 - MIFI_VECTOR_RESIZE, 297
 - PI, 297
- isAxisType
 - MetNoFimex::CoordinateAxis, 101
- isComplete
 - MetNoFimex::CoordinateSystem, 103
- isCSFor
 - MetNoFimex::CoordinateSystem, 104
- isDegree
 - MetNoFimex::Projection, 195
 - MetNoFimex::ProjectionImpl, 198
- isEnabledFor
 - MetNoFimex::Logger, 180
- isExplicit
 - MetNoFimex::CoordinateAxis, 101
- isLogging
 - felt::FeltFile, 147
- isSimpleSpatialGridded
 - MetNoFimex::CoordinateSystem, 104
- isSpatialVector
 - MetNoFimex::CDMVariable, 95
- isTime
 - MetNoFimex::Units, 223
- isUndefined
 - felt, 28
- isUnlimited
 - MetNoFimex::CDMDimension, 70
- isValid
 - MetNoFimex::GribFileMessage, 166
- iterator
 - felt::FeltFile, 146
- join
 - MetNoFimex, 39
- joinPtr
 - MetNoFimex, 39
- LambertConformalConicProjection
 - MetNoFimex::LambertConformalConicProjection, 176
- lastTime
 - felt::FeltFile, 146
- lastUpdateTime
 - felt::FeltFile, 146
- Lat
 - MetNoFimex::CoordinateAxis, 101
- LatitudeLongitudeProjection
 - MetNoFimex::LatitudeLongitudeProjection, 178
- LeftLowerHorizontal
 - felt::FeltGridDefinition, 148
 - MetNoFimex::GridDefinition, 172
- LeftLowerHorizontalAlternating
 - MetNoFimex::GridDefinition, 172
- LeftLowerVertical
 - MetNoFimex::GridDefinition, 172
- LeftLowerVerticalAlternating
 - MetNoFimex::GridDefinition, 172
- LeftUpperHorizontal
 - felt::FeltGridDefinition, 148
 - MetNoFimex::GridDefinition, 172

- LeftUpperHorizontalAlternating
 - MetNoFimex::GridDefinition, 172
- LeftUpperVertical
 - MetNoFimex::GridDefinition, 172
- LeftUpperVerticalAlternating
 - MetNoFimex::GridDefinition, 172
- level1
 - felt::FeltField, 143
- level2
 - felt::FeltField, 143
- LevelPair
 - MetNoFelt, 30
- listCoordinateSystems
 - MetNoFimex, 39
- listFeltArrays
 - MetNoFelt::Felt_File, 133
 - MetNoFelt::Felt_File2, 137
- listMessages
 - MetNoFimex::GribFileIndex, 165
- log
 - felt::FeltFile, 146
- LOG4FIMEX
 - Logger.h, 304
- Logger
 - MetNoFimex::Logger, 180
- logger
 - MetNoFimex::GribApiCDMWriter_ImplAbstract, 163
- Logger.h
 - LOG4FIMEX, 304
- LoggerPtr
 - MetNoFimex, 35
- LogLevel
 - MetNoFimex::Logger, 180
- Lon
 - MetNoFimex::CoordinateAxis, 101
- M_PI
 - FeltConstants.h, 228
- mday
 - MetNoFimex::FimexTime, 154
- MercatorProjection
 - MetNoFimex::MercatorProjection, 182
- MetNoFelt, 30
 - ANY_ARRAY, 31
 - ANY_ARRAY20, 31
 - ANY_VALUE, 31
 - getProjString, 31
 - index16toLevelPair, 31
 - index16toTime, 31
 - LevelPair, 30
 - ShortPairMap, 30
 - ShortPairSet, 30
 - UNDEFINED, 31
- MetNoFelt::Felt_Array, 122
 - ~Felt_Array, 123
 - addIdent19, 125
 - addInformationByIndex, 123
 - Felt_Array, 123
 - getDataHeader, 124
 - getDatatype, 124
 - getFieldSize, 126
 - getFillValue, 124
 - getGridParameters, 124
 - getGridType, 124
 - getIdent19, 125
 - getIndex, 126
 - getIndexHeader, 123
 - getLevelPairs, 125
 - getLevels, 125
 - getLevelType, 124
 - getName, 124
 - getScalingFactor, 126
 - getTimes, 125
 - getVerticalFeltType, 126
 - getX, 126
 - getY, 126
 - setDataHeader, 123
 - setFillValue, 124
 - setGridParameters, 124
 - setGridType, 124
- MetNoFelt::Felt_Array2, 127
 - ~Felt_Array2, 128
 - addInformationByField, 128
 - Felt_Array2, 127
 - getDatatype, 128
 - getField, 130
 - getFillValue, 129
 - getGrid, 128
 - getGridAllowDelta, 128
 - getGridDefinition, 129
 - getGridType, 130
 - getIdent19, 129
 - getLevelPairs, 129
 - getLevelType, 128
 - getName, 128
 - getScalingFactor, 129
 - getTimes, 129
 - getX, 129
 - getY, 129
 - scaleFactor, 129
- MetNoFelt::Felt_File, 131
 - ~Felt_File, 132
 - Felt_File, 132
 - getDataSlice, 132
 - getFeltArray, 132
 - getFeltLevelPairs, 133
 - getFeltLevels, 133

- getFeltTimes, 133
- getGridParameters, 134
- getGridType, 134
- getHybridLevels, 133
- getNX, 133
- getNY, 134
- getScaledDataSlice, 133
- getXData, 134
- getYData, 134
- listFeltArrays, 133
- MetNoFelt::Felt_File2, 135
 - ~Felt_File2, 136
 - Felt_File2, 136
 - getFeltArray, 136
 - getFeltLevelPairs, 137
 - getFeltTimes, 137
 - getGridDefinition, 138
 - getGridType, 137
 - getHybridLevels, 137
 - getNX, 137
 - getNY, 137
 - getScaledDataSlice, 136
 - getXData, 137
 - getYData, 137
 - listFeltArrays, 137
- MetNoFelt::Felt_File_Error, 139
 - Felt_File_Error, 139
- MetNoFelt::FeltParameters, 150
 - ~FeltParameters, 150
 - DEFAULT_CONFIG, 151
 - FeltParameters, 150
 - getParameterDatatype, 151
 - getParameterFillValue, 151
 - getParameterName, 151
 - getParameters, 151
- MetNoFelt::LevelPairLess, 179
 - operator(), 179
- MetNoFelt::ShortPairLess, 208
 - operator(), 208
- MetNoFimex, 32
 - CDM_CHAR, 35
 - CDM_DOUBLE, 36
 - CDM_FLOAT, 36
 - CDM_INT, 36
 - CDM_NAT, 35
 - CDM_SHORT, 36
 - CDM_STRING, 36
 - CDMDataType, 35
 - constConvertArrayType, 36
 - convertArrayType, 36
 - createData, 36
 - createDataSlice, 37
 - datatype2string, 37
 - defaultLogLevel, 37
 - DEPRECATED, 37
 - duplicateArrayType, 38
 - epoch_seconds, 35
 - getFeltReader, 38
 - getLogger, 38
 - getXmlName, 38
 - getXmlProp, 38
 - gribDataRead, 38
 - gribGetGridOrientation, 39
 - handleUdUnitError, 39
 - join, 39
 - joinPtr, 39
 - listCoordinateSystems, 39
 - LoggerPtr, 35
 - operator<<, 40
 - posixTime2epochTime, 40
 - recursiveCopyMultiDimData, 40
 - round, 41
 - string2datatype, 41
 - string2FimexTime, 41
 - string2lowerCase, 41
 - string2type, 41
 - tokenize, 41
 - tokenizeDotted, 42
 - trim, 42
 - type2string, 42
 - type2string< double >, 42
 - XPathObjPtr, 35
- MetNoFimex::C_CDMReader, 45
 - ~C_CDMReader, 45
 - C_CDMReader, 45
 - getDataSlice, 45
 - setDoubleCallbackFunction, 46
- MetNoFimex::CachedForwardInterpolation, 47
 - ~CachedForwardInterpolation, 47
 - CachedForwardInterpolation, 47
 - interpolateValues, 47
- MetNoFimex::CachedInterpolation, 48
 - ~CachedInterpolation, 48
 - CachedInterpolation, 48
 - interpolateValues, 49
- MetNoFimex::CachedInterpolationInterface, 50
 - interpolateValues, 50
- MetNoFimex::CachedVectorReprojection, 51
 - ~CachedVectorReprojection, 51
 - CachedVectorReprojection, 51
 - getXSize, 51
 - getYSize, 51
 - reprojectValues, 51
- MetNoFimex::CDM, 52
 - ~CDM, 55
 - addAttribute, 59
 - addDimension, 57
 - addOrReplaceAttribute, 59

- addVariable, 55
- AttrVec, 55
- CDM, 55
- checkVariableAttribute, 57
- DEPRECATED, 62
- DimVec, 55
- findVariables, 56
- generateProjectionCoordinates, 62
- getAttribute, 60, 61
- getAttributes, 60
- getDimension, 58
- getDimensions, 60
- getFillValue, 61
- getHorizontalXAxis, 63
- getHorizontalYAxis, 63
- getLatitudeLongitude, 63
- getProjectionOf, 63
- getTimeAxis, 64
- getUnits, 61
- getUnlimitedDim, 59
- getVariable, 55
- getVariables, 60
- getVerticalAxis, 64
- globalAttributeNS, 60
- hasDimension, 57
- hasUnlimitedDim, 59
- hasVariable, 56
- operator=, 55
- removeAttribute, 59
- removeDimension, 58
- removeVariable, 57
- renameDimension, 58
- renameVariable, 56
- StrAttrVecMap, 55
- testDimensionInUse, 58
- toXMLStream, 60
- VarVec, 55
- MetNoFimex::CDMAttribute, 65
 - ~CDMAttribute, 67
 - CDMAttribute, 66, 67
 - getData, 67
 - getDataType, 67
 - getName, 67
 - getStringValue, 67
 - setData, 67
 - setName, 67
 - toXMLStream, 67
- MetNoFimex::CDMDimension, 69
 - ~CDMDimension, 69
 - CDMDimension, 69
 - getLength, 70
 - getName, 69
 - isUnlimited, 70
 - setLength, 70
 - setName, 69
 - setUnlimited, 70
 - toXMLStream, 70
- MetNoFimex::CDMException, 71
 - CDMException, 71
- MetNoFimex::CDMExtractor, 72
 - ~CDMExtractor, 72
 - CDMExtractor, 72
 - changeDataType, 74
 - getDataSlice, 72
 - reduceDimension, 73
 - reduceDimensionStartEnd, 73
 - removeVariable, 73
- MetNoFimex::CDMFileReaderFactory, 75
 - create, 75
 - detectFileType, 75
- MetNoFimex::CDMInterpolator, 77
 - ~CDMInterpolator, 77
 - CDMInterpolator, 77
 - changeProjection, 77, 78
 - getDataSlice, 77
 - getLatitudeName, 78
 - getLongitudeName, 79
 - setLatitudeName, 78
 - setLongitudeName, 78
- MetNoFimex::CDMNameCompare, 80
 - operator(), 80
- MetNoFimex::CDMNamedEntity, 81
 - ~CDMNamedEntity, 81
 - getName, 81
- MetNoFimex::CDMNameEqual, 82
 - ~CDMNameEqual, 82
 - CDMNameEqual, 82
 - operator(), 82
- MetNoFimex::CDMNameEqualPtr, 83
 - ~CDMNameEqualPtr, 83
 - CDMNameEqualPtr, 83
 - operator(), 83
- MetNoFimex::CDMQualityExtractor, 84
 - ~CDMQualityExtractor, 85
 - CDMQualityExtractor, 84
 - getDataSlice, 85
 - getStatusVariable, 85
 - getVariableFlags, 85
 - getVariableValues, 85
- MetNoFimex::CDMReader, 86
 - ~CDMReader, 88
 - cdm_, 91
 - CDMReader, 88
 - getCDM, 88
 - getData, 88
 - getDataSlice, 88
 - getDataSliceFromMemory, 90
 - getScaledData, 90

- getScaledDataInUnit, 90
- getScaledDataSlice, 89
- getScaledDataSliceInUnit, 89, 90
- MetNoFimex::CDMTimeInterpolator, 92
 - ~CDMTimeInterpolator, 92
 - CDMTimeInterpolator, 92
 - changeTimeAxis, 92
 - getDataSlice, 92
- MetNoFimex::CDMVariable, 94
 - ~CDMVariable, 95
 - CDMVariable, 95
 - checkDimension, 95
 - getData, 96
 - getDataType, 95
 - getName, 95
 - getShape, 95
 - getSpatialVectorCounterpart, 95
 - getSpatialVectorDirection, 95
 - hasData, 96
 - isSpatialVector, 95
 - setAsSpatialVector, 95
 - setData, 96
 - setDataType, 95
 - setName, 95
 - setShape, 95
 - toXMLStream, 96
- MetNoFimex::CDMWriter, 97
 - ~CDMWriter, 97
 - cdmReader, 97
 - CDMWriter, 97
 - outputFile, 97
- MetNoFimex::ChangeMissingValue, 98
 - ChangeMissingValue, 98
 - operator(), 98
- MetNoFimex::CompleteCoordinateSystemForComparator, 99
 - ~CompleteCoordinateSystemForComparator, 99
 - CompleteCoordinateSystemForComparator, 99
 - operator(), 99
- MetNoFimex::CoordinateAxis, 100
 - ~CoordinateAxis, 101
 - AxisType, 100
 - CoordinateAxis, 101
 - GeoX, 100
 - GeoY, 100
 - GeoZ, 100
 - getAxisType, 101
 - getAxisTypeStr, 101
 - Height, 101
 - isAxisType, 101
 - isExplicit, 101
 - Lat, 101
 - Lon, 101
 - operator<, 101
 - Pressure, 101
 - setAxisType, 101
 - setExplicit, 101
 - Time, 100
 - type2string, 101
 - Undefined, 100
- MetNoFimex::CoordinateSystem, 102
 - ~CoordinateSystem, 103
 - AxisPtr, 102
 - ConstAxisList, 103
 - ConstAxisPtr, 102
 - CoordinateSystem, 103
 - findAxisOfType, 105
 - getAxes, 106
 - getConventionName, 103
 - getGeoXAxis, 105
 - getGeoYAxis, 105
 - getGeoZAxis, 105
 - getProjection, 104
 - getTimeAxis, 106
 - hasAxisType, 104
 - hasProjection, 104
 - id, 103
 - isComplete, 103
 - isCSFor, 104
 - isSimpleSpatialGridded, 104
 - setAxis, 106
 - setComplete, 103
 - setConventionName, 103
 - setCSFor, 104
 - setProjection, 104
 - setSimpleSpatialGridded, 104
- MetNoFimex::Data, 107
 - ~Data, 108
 - asChar, 109
 - asConstChar, 109
 - asConstDouble, 110
 - asConstFloat, 109
 - asConstInt, 109
 - asConstShort, 109
 - asDouble, 110
 - asFloat, 110
 - asInt, 109
 - asShort, 109
 - asString, 110
 - bytes_for_one, 108
 - convertDataType, 111
 - getDataPtr, 108
 - getDataType, 111
 - setAllValues, 110
 - setValue, 110
 - setValues, 110

- size, 108
- slice, 111
- toStream, 108
- MetNoFimex::DataImpl, 112
 - ~DataImpl, 114
 - as, 114, 115
 - asBase, 114
 - asChar, 115
 - asConstChar, 115
 - asConstDouble, 116
 - asConstFloat, 115
 - asConstInt, 115
 - asConstShort, 115
 - asDouble, 116
 - asFloat, 116
 - asInt, 115
 - asShort, 115
 - asString, 116
 - bytes_for_one, 114
 - convertDataType, 117
 - DataImpl, 114
 - getDataPtr, 114
 - getDataType, 118
 - setAllValues, 117
 - setValue, 116
 - setValues, 116, 118, 119
 - size, 114
 - slice, 117
 - toStream, 114
- MetNoFimex::DataTypeChanger, 120
 - ~DataTypeChanger, 120
 - convertData, 121
 - DataTypeChanger, 120
 - getDataType, 121
- MetNoFimex::FeltCDMReader, 140
 - ~FeltCDMReader, 140
 - FeltCDMReader, 140
 - getDataSlice, 140
- MetNoFimex::FeltCDMReader2, 141
 - ~FeltCDMReader2, 141
 - FeltCDMReader2, 141
 - getDataSlice, 141
- MetNoFimex::FimexTime, 152
 - hour, 154
 - mday, 154
 - minute, 153
 - month, 154
 - msecond, 153
 - operator!=, 153
 - operator<, 153
 - operator<=, 153
 - operator>, 153
 - operator>=, 153
 - operator==, 153
 - second, 153
 - year, 154
- MetNoFimex::GribApiCDMWriter, 155
 - ~GribApiCDMWriter, 155
 - GribApiCDMWriter, 155
- MetNoFimex::GribApiCDMWriter_Impl1, 156
 - ~GribApiCDMWriter_Impl1, 156
 - GribApiCDMWriter_Impl1, 156
 - handleTypeScaleAndMissingData, 157
 - setLevel, 157
 - setParameter, 156
 - setProjection, 156
- MetNoFimex::GribApiCDMWriter_Impl2, 158
 - ~GribApiCDMWriter_Impl2, 158
 - GribApiCDMWriter_Impl2, 158
 - handleTypeScaleAndMissingData, 159
 - setLevel, 159
 - setParameter, 158
 - setProjection, 158
- MetNoFimex::GribApiCDMWriter_ImplAbstract, 160
 - ~GribApiCDMWriter_ImplAbstract, 161
 - configFile, 163
 - getLevels, 162
 - getNodePtr, 162
 - getTimes, 162
 - GribApiCDMWriter_ImplAbstract, 161
 - gribHandle, 163
 - gribVersion, 163
 - handleTypeScaleAndMissingData, 162
 - logger, 163
 - run, 161
 - setData, 161
 - setGlobalAttributes, 161
 - setLevel, 162
 - setParameter, 161
 - setProjection, 161
 - setTime, 161
 - writeGribHandleToFile, 162
 - xmlConfig, 163
- MetNoFimex::GribCDMReader, 164
 - ~GribCDMReader, 164
 - getDataSlice, 164
 - GribCDMReader, 164
- MetNoFimex::GribFileIndex, 165
 - ~GribFileIndex, 165
 - getUrl, 165
 - GribFileIndex, 165
 - listMessages, 165
- MetNoFimex::GribFileMessage, 166
 - ~GribFileMessage, 166
 - getDateTime, 167
 - getEdition, 167
 - getFilePosition, 167

- getFileURL, 167
- getGridDefinition, 167
- getLevelNumber, 167
- getLevelType, 167
- getMessageNumber, 167
- getName, 167
- getParameterIds, 167
- getShortName, 167
- getTypeOfGrid, 167
- GribFileMessage, 166
- isValid, 166
- toString, 166
- MetNoFimex::GribFileMessageEqualLevelTime, 169
 - ~GribFileMessageEqualLevelTime, 169
 - GribFileMessageEqualLevelTime, 169
 - operator(), 169
- MetNoFimex::GribFileMessageEqualTime, 170
 - ~GribFileMessageEqualTime, 170
 - GribFileMessageEqualTime, 170
 - operator(), 170
- MetNoFimex::GridDefinition, 171
 - ~GridDefinition, 173
 - comparableTo, 174
 - getProjDefinition, 173
 - getScanMode, 174
 - getXIncrement, 173
 - getXSize, 173
 - getXStart, 173
 - getYIncrement, 173
 - getYSize, 173
 - getYStart, 174
 - GridDefinition, 173
 - LeftLowerHorizontal, 172
 - LeftLowerHorizontalAlternating, 172
 - LeftLowerVertical, 172
 - LeftLowerVerticalAlternating, 172
 - LeftUpperHorizontal, 172
 - LeftUpperHorizontalAlternating, 172
 - LeftUpperVertical, 172
 - LeftUpperVerticalAlternating, 172
 - Orientation, 172
 - OrientationFlags, 172
 - RightLowerHorizontal, 172
 - RightLowerHorizontalAlternating, 172
 - RightLowerVertical, 172
 - RightLowerVerticalAlternating, 172
 - RightUpperHorizontal, 172
 - RightUpperHorizontalAlternating, 172
 - RightUpperVertical, 172
 - RightUpperVerticalAlternating, 172
 - ScanIsAlternating, 172
 - ScanIsVertical, 172
 - ScanStartBottom, 172
 - ScanStartRight, 172
 - setProjDefinition, 173
 - setScanMode, 174
 - setXIncrement, 173
 - setXSize, 173
 - setXStart, 173
 - setYIncrement, 173
 - setYSize, 173
 - setYStart, 174
- MetNoFimex::LambertConformalConicProjection, 175
 - ~LambertConformalConicProjection, 176
 - acceptsProj4, 176
 - getProj4ProjectionPart, 176
 - LambertConformalConicProjection, 176
 - parametersFromProj4, 176
- MetNoFimex::LatitudeLongitudeProjection, 177
 - ~LatitudeLongitudeProjection, 178
 - acceptsProj4, 178
 - getProj4ProjectionPart, 178
 - LatitudeLongitudeProjection, 178
 - parametersFromProj4, 178
- MetNoFimex::Logger, 180
 - ~Logger, 180
 - DEBUG, 180
 - ERROR, 180
 - FATAL, 180
 - forcedLog, 180
 - INFO, 180
 - isEnabledFor, 180
 - Logger, 180
 - LogLevel, 180
 - OFF, 180
 - WARN, 180
- MetNoFimex::MercatorProjection, 182
 - ~MercatorProjection, 182
 - acceptsProj4, 182
 - getProj4ProjectionPart, 182
 - MercatorProjection, 182
 - parametersFromProj4, 182
- MetNoFimex::NcmlCDMReader, 185
 - ~NcmlCDMReader, 186
 - getDataSlice, 186
 - NcmlCDMReader, 185
- MetNoFimex::NetCDF_CDMReader, 187
 - ~NetCDF_CDMReader, 187
 - getDataSlice, 187, 188
 - NetCDF_CDMReader, 187
- MetNoFimex::NetCDF_CDMWriter, 189
 - ~NetCDF_CDMWriter, 189
 - getAttribute, 190
 - getDimensionName, 189
 - getVariableName, 189
 - NetCDF_CDMWriter, 189

- MetNoFimex::Null_CDMWriter, 191
 - ~Null_CDMWriter, 191
 - Null_CDMWriter, 191
- MetNoFimex::PolarStereographicProjection, 192
 - ~PolarStereographicProjection, 192
 - acceptsProj4, 192
 - parametersFromProj4, 192
 - PolarStereographicProjection, 192
- MetNoFimex::Projection, 194
 - ~Projection, 195
 - addParameter, 195
 - addParameters, 195
 - create, 196
 - createByProj4, 196
 - getName, 195
 - getParameters, 195
 - getProj4String, 195
 - isDegree, 195
 - operator==, 196
 - Projection, 195
 - removeParameter, 195
 - toString, 195
- MetNoFimex::ProjectionImpl, 197
 - ~ProjectionImpl, 198
 - addParameter, 198
 - addParameters, 198
 - addParameterToStream, 199
 - getName, 198
 - getParameters, 198
 - getProj4ProjectionPart, 199
 - getProj4String, 198
 - isDegree, 198
 - params_, 200
 - proj4GetEarthAttributes, 199
 - proj4ProjectionMatchesName, 199
 - ProjectionImpl, 198
 - removeParameter, 198
 - toString, 199
- MetNoFimex::ReplaceStringObject, 201
 - ~ReplaceStringObject, 201
 - put, 201
 - setFormatString, 201
 - setFormatStringAndOptions, 201
- MetNoFimex::ReplaceStringTimeObject, 203
 - ~ReplaceStringTimeObject, 203
 - operator<<, 204
 - put, 203
 - ReplaceStringTimeObject, 203
 - setFormatString, 203
 - setFormatStringAndOptions, 204
- MetNoFimex::RotatedLatitudeLongitudeProjection, 205
 - ~RotatedLatitudeLongitudeProjection, 206
 - acceptsProj4, 206
 - getProj4ProjectionPart, 206
 - parametersFromProj4, 206
 - RotatedLatitudeLongitudeProjection, 206
- MetNoFimex::ScaleValue, 207
 - operator(), 207
 - ScaleValue, 207
- MetNoFimex::SliceBuilder, 209
 - ~SliceBuilder, 209
 - getDimensionNames, 210
 - getDimensionSizes, 210
 - getDimensionStartPositions, 210
 - getMaxDimensionSizes, 210
 - setStartAndSize, 209
 - SliceBuilder, 209
- MetNoFimex::SpatialAxisSpec, 211
 - ~SpatialAxisSpec, 212
 - getAxisSteps, 212
 - requireStartEnd, 212
 - setStartEnd, 212
 - SpatialAxisSpec, 211
- MetNoFimex::staticCast, 213
 - operator(), 213
- MetNoFimex::StereographicProjection, 214
 - ~StereographicProjection, 214
 - acceptsProj4, 214
 - getProj4ProjectionPart, 215
 - parametersFromProj4, 214
 - StereographicProjection, 214
- MetNoFimex::TimeLevelDataSliceFetcher, 216
 - ~TimeLevelDataSliceFetcher, 216
 - getTimeLevelSlice, 216
 - TimeLevelDataSliceFetcher, 216
- MetNoFimex::TimeSpec, 217
 - ~TimeSpec, 218
 - getTimeSteps, 218
 - getUnitString, 218
 - TimeSpec, 217
- MetNoFimex::TimeUnit, 219
 - ~TimeUnit, 220
 - epochSeconds2unitTime, 220
 - fimexTime2unitTime, 220
 - fimexTime2unitTimeX, 220
 - TimeUnit, 219
 - unitTime2epochSeconds, 220
 - unitTime2fimexTime, 220
- MetNoFimex::UnitException, 221
 - UnitException, 221
- MetNoFimex::Units, 222
 - ~Units, 222
 - areConvertible, 222
 - convert, 222
 - isTime, 223
 - operator=, 222
 - Units, 222

- MetNoFimex::XMLDoc, 224
 - ~XMLDoc, 224
 - getXPathObject, 224
 - registerNamespace, 225
 - XMLDoc, 224
- mifi_3d_array_position
 - interpolation.h, 297
- mifi_bad2nanf
 - interpolation.h, 298
- mifi_cdm_reader, 184
 - c_fimex.h, 236
 - get, 184
 - mifi_cdm_reader, 184
 - mifi_cdm_reader, 184
- MIFI_DEBUG
 - interpolation.h, 295
- MIFI_EARTH_RADIUS_M
 - CDMconstants.h, 247
- MIFI_ERROR
 - interpolation.h, 295
- MIFI_FILETYPE_FELT
 - CDMconstants.h, 247
- MIFI_FILETYPE_GRIB
 - CDMconstants.h, 247
- MIFI_FILETYPE_NCML
 - CDMconstants.h, 247
- MIFI_FILETYPE_NETCDF
 - CDMconstants.h, 247
- MIFI_FILETYPE_UNKNOWN
 - CDMconstants.h, 247
- mifi_free_cdm_reader
 - c_fimex.h, 237
- mifi_get_double_data
 - c_fimex.h, 237
- mifi_get_double_datasize
 - c_fimex.h, 237
- mifi_get_values_bicubic_f
 - interpolation.h, 298
- mifi_get_values_bilinear_f
 - interpolation.h, 298
- mifi_get_values_f
 - interpolation.h, 299
- mifi_get_values_linear_f
 - interpolation.h, 299
- mifi_get_variable_name
 - c_fimex.h, 237
- mifi_get_variable_number
 - c_fimex.h, 238
- mifi_get_vector_reproject_matrix
 - interpolation.h, 299
- MIFI_GRIB_CHECK
 - GribUtils.h, 291
- mifi_grib_check
 - GribUtils.h, 291
- mifi_grib_writer
 - c_fimex.h, 238
- MIFI_INTERPOL_BICUBIC
 - interpolation.h, 295
- MIFI_INTERPOL_BILINEAR
 - interpolation.h, 295
- MIFI_INTERPOL_COORD_NN
 - interpolation.h, 295
- MIFI_INTERPOL_COORD_NN_KD
 - interpolation.h, 296
- MIFI_INTERPOL_FORWARD_MAX
 - interpolation.h, 296
- MIFI_INTERPOL_FORWARD_MEAN
 - interpolation.h, 296
- MIFI_INTERPOL_FORWARD_MEDIAN
 - interpolation.h, 296
- MIFI_INTERPOL_FORWARD_MIN
 - interpolation.h, 296
- MIFI_INTERPOL_FORWARD_SUM
 - interpolation.h, 296
- MIFI_INTERPOL_NEAREST_NEIGHBOR
 - interpolation.h, 296
- mifi_interpolate_d
 - interpolation.h, 300
- mifi_interpolate_f
 - interpolation.h, 300
- MIFI_LATITUDE
 - interpolation.h, 297
- MIFI_LONGITUDE
 - interpolation.h, 297
- mifi_nanf2bad
 - interpolation.h, 301
- mifi_netcdf_writer
 - c_fimex.h, 238
- mifi_new_c_reader
 - c_fimex.h, 238
- mifi_new_cdminterpolator
 - c_fimex.h, 239
- mifi_new_felt_reader
 - c_fimex.h, 239
- mifi_new_grib_reader
 - c_fimex.h, 239
- mifi_new_ncml_modifier
 - c_fimex.h, 240
- mifi_new_ncml_reader
 - c_fimex.h, 240
- mifi_new_netcdf_reader
 - c_fimex.h, 240
- mifi_nullcdm_writer
 - c_fimex.h, 240
- MIFI_OK
 - interpolation.h, 297
- mifi_points2position
 - interpolation.h, 301

- MIFI_PROJ_AXIS
 - interpolation.h, 297
- mifi_project_axes
 - interpolation.h, 301
- mifi_project_values
 - interpolation.h, 302
- mifi_set_callback_double
 - c_fimex.h, 240
- MIFI_UNDEFINED_D
 - interpolation.h, 297
- MIFI_UNDEFINED_F
 - interpolation.h, 297
- MIFI_VECTOR_KEEP_SIZE
 - interpolation.h, 297
- mifi_vector_reproject_values_by_matrix_f
 - interpolation.h, 302
- mifi_vector_reproject_values_f
 - interpolation.h, 303
- MIFI_VECTOR_RESIZE
 - interpolation.h, 297
- minute
 - MetNoFimex::FimexTime, 153
- miscField
 - felt::FeltField, 144
- month
 - MetNoFimex::FimexTime, 154
- msecond
 - MetNoFimex::FimexTime, 153
- NcmlCDMReader
 - MetNoFimex::NcmlCDMReader, 185
- NetCDF_CDMReader
 - MetNoFimex::NetCDF_CDMReader, 187
- NetCDF_CDMWriter
 - MetNoFimex::NetCDF_CDMWriter, 189
- Null_CDMWriter
 - MetNoFimex::Null_CDMWriter, 191
- OFF
 - MetNoFimex::Logger, 180
- offsetToContentDefinition
 - felt, 29
- operator!=
 - MetNoFimex::FimexTime, 153
- operator<
 - MetNoFimex::CoordinateAxis, 101
 - MetNoFimex::FimexTime, 153
- operator<<
 - MetNoFimex, 40
 - MetNoFimex::ReplaceStringTimeObject, 204
- operator<=
 - MetNoFimex::FimexTime, 153
- operator>
 - MetNoFimex::FimexTime, 153
- operator>=
 - MetNoFimex::FimexTime, 153
- operator()
 - MetNoFelt::LevelPairLess, 179
 - MetNoFelt::ShortPairLess, 208
 - MetNoFimex::CDMNameCompare, 80
 - MetNoFimex::CDMNameEqual, 82
 - MetNoFimex::CDMNameEqualPtr, 83
 - MetNoFimex::ChangeMissingValue, 98
 - MetNoFimex::CompleteCoordinateSystemForComparator, 99
 - MetNoFimex::GribFileMessageEqualLevelTime, 169
 - MetNoFimex::GribFileMessageEqualTime, 170
 - MetNoFimex::ScaleValue, 207
 - MetNoFimex::staticCast, 213
- operator=
 - MetNoFimex::CDM, 55
 - MetNoFimex::Units, 222
- operator==
 - MetNoFimex::FimexTime, 153
 - MetNoFimex::Projection, 196
- Orientation
 - felt::FeltGridDefinition, 148
 - MetNoFimex::GridDefinition, 172
- OrientationFlags
 - MetNoFimex::GridDefinition, 172
- outputFile
 - MetNoFimex::CDMWriter, 97
- parameter
 - felt::FeltField, 143
- parametersFromProj4
 - MetNoFimex::LambertConformalConicProjection, 176
 - MetNoFimex::LatitudeLongitudeProjection, 178
 - MetNoFimex::MercatorProjection, 182
 - MetNoFimex::PolarStereographicProjection, 192
 - MetNoFimex::RotatedLatitudeLongitudeProjection, 206
 - MetNoFimex::StereographicProjection, 214
- params_
 - MetNoFimex::ProjectionImpl, 200
- parseTime
 - felt, 28
- parseTimeNoThrow
 - felt, 29
- PI
 - felt, 29
 - interpolation.h, 297
- PolarStereographicProjection

- MetNoFimex::PolarStereographicProjection, 192
- posixTime2epochTime
 - MetNoFimex, 40
- Pressure
 - MetNoFimex::CoordinateAxis, 101
- producer
 - felt::FeltField, 143
- proj4GetEarthAttributes
 - MetNoFimex::ProjectionImpl, 199
- proj4ProjectionMatchesName
 - MetNoFimex::ProjectionImpl, 199
- projDefinition
 - felt::FeltGridDefinition, 149
- Projection
 - MetNoFimex::Projection, 195
- ProjectionImpl
 - MetNoFimex::ProjectionImpl, 198
- projectionInformation
 - felt::FeltField, 144
- put
 - MetNoFimex::ReplaceStringObject, 201
 - MetNoFimex::ReplaceStringTimeObject, 203
- recursiveCopyMultiDimData
 - MetNoFimex, 40
- reduceDimension
 - MetNoFimex::CDMExtractor, 73
- reduceDimensionStartEnd
 - MetNoFimex::CDMExtractor, 73
- referenceTime
 - felt::FeltField, 143
 - felt::FeltFile, 146
- registerNamespace
 - MetNoFimex::XMLDoc, 225
- removeAttribute
 - MetNoFimex::CDM, 59
- removeDimension
 - MetNoFimex::CDM, 58
- removeParameter
 - MetNoFimex::Projection, 195
 - MetNoFimex::ProjectionImpl, 198
- removeVariable
 - MetNoFimex::CDM, 57
 - MetNoFimex::CDMExtractor, 73
- renameDimension
 - MetNoFimex::CDM, 58
- renameVariable
 - MetNoFimex::CDM, 56
- ReplaceStringTimeObject
 - MetNoFimex::ReplaceStringTimeObject, 203
- reprojectValues
 - MetNoFimex::CachedVectorReprojection, 51
- requireStartEnd
 - MetNoFimex::SpatialAxisSpec, 212
- RightLowerHorizontal
 - MetNoFimex::GridDefinition, 172
- RightLowerHorizontalAlternating
 - MetNoFimex::GridDefinition, 172
- RightLowerVertical
 - MetNoFimex::GridDefinition, 172
- RightLowerVerticalAlternating
 - MetNoFimex::GridDefinition, 172
- RightUpperHorizontal
 - MetNoFimex::GridDefinition, 172
- RightUpperHorizontalAlternating
 - MetNoFimex::GridDefinition, 172
- RightUpperVertical
 - MetNoFimex::GridDefinition, 172
- RightUpperVerticalAlternating
 - MetNoFimex::GridDefinition, 172
- RotatedLatitudeLongitudeProjection
 - MetNoFimex::RotatedLatitudeLongitudeProjection, 206
- round
 - MetNoFimex, 41
- run
 - MetNoFimex::GribApiCDMWriter_ - ImplAbstract, 161
- scaleFactor
 - felt::FeltField, 143
 - MetNoFimex::Felt_Array2, 129
- ScaleValue
 - MetNoFimex::ScaleValue, 207
- ScanIsAlternating
 - MetNoFimex::GridDefinition, 172
- ScanIsVertical
 - MetNoFimex::GridDefinition, 172
- ScanStartBottom
 - MetNoFimex::GridDefinition, 172
- ScanStartRight
 - MetNoFimex::GridDefinition, 172
- second
 - MetNoFimex::FimexTime, 153
- setAllValues
 - MetNoFimex::Data, 110
 - MetNoFimex::DataImpl, 117
- setAsSpatialVector
 - MetNoFimex::CDMVariable, 95
- setAxis
 - MetNoFimex::CoordinateSystem, 106
- setAxisType
 - MetNoFimex::CoordinateAxis, 101
- setComplete
 - MetNoFimex::CoordinateSystem, 103
- setConventionName
 - MetNoFimex::CoordinateSystem, 103

- setCSFor
 - MetNoFimex::CoordinateSystem, 104
- setData
 - MetNoFimex::CDMAttribute, 67
 - MetNoFimex::CDMVariable, 96
 - MetNoFimex::GribApiCDMWriter_ - ImplAbstract, 161
- setDataHeader
 - MetNoFelt::Felt_Array, 123
- setDataType
 - MetNoFimex::CDMVariable, 95
- setDoubleCallbackFunction
 - MetNoFimex::C_CDMReader, 46
- setExplicit
 - MetNoFimex::CoordinateAxis, 101
- setFillValue
 - MetNoFelt::Felt_Array, 124
- setFormatString
 - MetNoFimex::ReplaceStringObject, 201
 - MetNoFimex::ReplaceStringTimeObject, 203
- setFormatStringAndOptions
 - MetNoFimex::ReplaceStringObject, 201
 - MetNoFimex::ReplaceStringTimeObject, 204
- setGlobalAttributes
 - MetNoFimex::GribApiCDMWriter_ - ImplAbstract, 161
- setGridParameters
 - MetNoFelt::Felt_Array, 124
- setGridType
 - MetNoFelt::Felt_Array, 124
- setLatitudeName
 - MetNoFimex::CDMInterpolator, 78
- setLength
 - MetNoFimex::CDMDimension, 70
- setLevel
 - MetNoFimex::GribApiCDMWriter_Impl1, 157
 - MetNoFimex::GribApiCDMWriter_Impl2, 159
 - MetNoFimex::GribApiCDMWriter_ - ImplAbstract, 162
- setLogging
 - felt::FeltFile, 147
- setLogStream
 - felt::FeltFile, 147
- setLongitudeName
 - MetNoFimex::CDMInterpolator, 78
- setName
 - MetNoFimex::CDMAttribute, 67
 - MetNoFimex::CDMDimension, 69
 - MetNoFimex::CDMVariable, 95
- setParameter
 - MetNoFimex::GribApiCDMWriter_Impl1, 156
 - MetNoFimex::GribApiCDMWriter_Impl2, 158
 - MetNoFimex::GribApiCDMWriter_ - ImplAbstract, 161
- setProjDefinition
 - MetNoFimex::GridDefinition, 173
- setProjection
 - MetNoFimex::CoordinateSystem, 104
 - MetNoFimex::GribApiCDMWriter_Impl1, 156
 - MetNoFimex::GribApiCDMWriter_Impl2, 158
 - MetNoFimex::GribApiCDMWriter_ - ImplAbstract, 161
- setScanMode
 - MetNoFimex::GridDefinition, 174
- setShape
 - MetNoFimex::CDMVariable, 95
- setSimpleSpatialGridded
 - MetNoFimex::CoordinateSystem, 104
- setStartAndSize
 - MetNoFimex::SliceBuilder, 209
- setStartEnd
 - MetNoFimex::SpatialAxisSpec, 212
- setTime
 - MetNoFimex::GribApiCDMWriter_ - ImplAbstract, 161
- setUnlimited
 - MetNoFimex::CDMDimension, 70
- setValue
 - MetNoFimex::Data, 110
 - MetNoFimex::DataImpl, 116
- setValues
 - MetNoFimex::Data, 110
 - MetNoFimex::DataImpl, 116, 118, 119
- setXIncrement
 - MetNoFimex::GridDefinition, 173
- setXSize
 - MetNoFimex::GridDefinition, 173
- setXStart
 - MetNoFimex::GridDefinition, 173
- setYIncrement
 - MetNoFimex::GridDefinition, 173
- setYSize
 - MetNoFimex::GridDefinition, 173
- setYStart
 - MetNoFimex::GridDefinition, 174
- ShortPairMap
 - MetNoFelt, 30
- ShortPairSet
 - MetNoFelt, 30
- size
 - felt::FeltFile, 146
 - MetNoFimex::Data, 108

- MetNoFimex::DataImpl, 114
- size_type
 - felt::FeltFile, 146
- slice
 - MetNoFimex::Data, 111
 - MetNoFimex::DataImpl, 117
- SliceBuilder
 - MetNoFimex::SliceBuilder, 209
- SpatialAxisSpec
 - MetNoFimex::SpatialAxisSpec, 211
- startLatitude
 - felt::FeltGridDefinition, 149
- startLongitude
 - felt::FeltGridDefinition, 149
- startX
 - felt::FeltGridDefinition, 149
- startY
 - felt::FeltGridDefinition, 149
- StereographicProjection
 - MetNoFimex::StereographicProjection, 214
- StrAttrVecMap
 - MetNoFimex::CDM, 55
- string2datatype
 - MetNoFimex, 41
- string2FimexTime
 - MetNoFimex, 41
- string2lowerCase
 - MetNoFimex, 41
- string2type
 - MetNoFimex, 41
- testDimensionInUse
 - MetNoFimex::CDM, 58
- Time
 - MetNoFimex::CoordinateAxis, 100
- TimeLevelDataSliceFetcher
 - MetNoFimex::TimeLevelDataSliceFetcher, 216
- TimeSpec
 - MetNoFimex::TimeSpec, 217
- TimeUnit
 - MetNoFimex::TimeUnit, 219
- tokenize
 - MetNoFimex, 41
- tokenizeDotted
 - MetNoFimex, 42
- toStream
 - MetNoFimex::Data, 108
 - MetNoFimex::DataImpl, 114
- toString
 - MetNoFimex::GribFileMessage, 166
 - MetNoFimex::Projection, 195
 - MetNoFimex::ProjectionImpl, 199
- toXMLStream
 - MetNoFimex::CDM, 60
 - MetNoFimex::CDMAAttribute, 67
 - MetNoFimex::CDMDimension, 70
 - MetNoFimex::CDMVariable, 96
- trim
 - MetNoFimex, 42
- type2string
 - MetNoFimex, 42
 - MetNoFimex::CoordinateAxis, 101
- type2string< double >
 - MetNoFimex, 42
- UNDEFINED
 - MetNoFelt, 31
- Undefined
 - MetNoFimex::CoordinateAxis, 100
- UnitException
 - MetNoFimex::UnitException, 221
- Units
 - MetNoFimex::Units, 222
- unitTime2epochSeconds
 - MetNoFimex::TimeUnit, 220
- unitTime2fimexTime
 - MetNoFimex::TimeUnit, 220
- valid
 - felt::FeltField, 143
- validTime
 - felt::FeltField, 143
- value
 - binary, 43
 - binary< 0 >, 44
- VarVec
 - MetNoFimex::CDM, 55
- verticalCoordinate
 - felt::FeltField, 143
- WARN
 - MetNoFimex::Logger, 180
- word
 - felt, 28
- writeGribHandleToFile
 - MetNoFimex::GribApiCDMWriter_ - ImplAbstract, 162
- xmlConfig
 - MetNoFimex::GribApiCDMWriter_ - ImplAbstract, 163
- XMLDoc
 - MetNoFimex::XMLDoc, 224
- xmlDoc
 - XMLDoc.h, 319
- XMLDoc.h
 - xmlDoc, 319

- [xmlNode, 319](#)
 - [xmlNodePtr, 319](#)
 - [xmlXPathContext, 319](#)
 - [xmlXPathObject, 319](#)
- [xmlNode](#)
 - [XMLDoc.h, 319](#)
- [xmlNodePtr](#)
 - [XMLDoc.h, 319](#)
- [xmlXPathContext](#)
 - [XMLDoc.h, 319](#)
- [xmlXPathObject](#)
 - [XMLDoc.h, 319](#)
- [xNum](#)
 - [felt::FeltField, 144](#)
- [XPathObjPtr](#)
 - [MetNoFimex, 35](#)
- [year](#)
 - [MetNoFimex::FimexTime, 154](#)
- [yNum](#)
 - [felt::FeltField, 144](#)