

MI - Fimex

Generated by Doxygen 1.5.5

Fri May 15 13:12:15 2009

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	23
4.1 ncml Configuration	24
5 gribWriter Configuration	27
5.1 gribWriter Configuration	28
6 netcdfWriter Configuration	31
6.1 netcdfWriter Configuration	32
7 Namespace Index	33
7.1 Namespace List	33
8 Class Index	35
8.1 Class Hierarchy	35
9 Class Index	37
9.1 Class List	37
10 File Index	39
10.1 File List	39
11 Namespace Documentation	41
11.1 KDTree Namespace Reference	41
11.2 MetNoFelt Namespace Reference	45

11.3 MetNoFimex Namespace Reference	47
12 Class Documentation	55
12.1 KDTree::_Alloc_base< _Tp, _Alloc > Class Template Reference	55
12.2 KDTree::_Alloc_base< _Tp, _Alloc >::NoLeakAlloc Class Reference	58
12.3 KDTree::_Base_iterator Class Reference	59
12.4 KDTree::_Bracket_accessor< _Val > Struct Template Reference	61
12.5 KDTree::_Iterator< _Val, _Ref, _Ptr > Class Template Reference	62
12.6 KDTree::_Node< _Val > Struct Template Reference	66
12.7 KDTree::_Node_base Struct Reference	67
12.8 KDTree::_Node_compare< _Val, _Acc, _Cmp > Class Template Reference	70
12.9 KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp > Struct Template Reference	71
12.10 KDTree::always_true< _Tp > Struct Template Reference	75
12.11 MetNoFimex::CachedInterpolation Class Reference	76
12.12 MetNoFimex::CachedVectorReprojection Class Reference	78
12.13 MetNoFimex::CDM Class Reference	79
12.14 MetNoFimex::CDMAttribute Class Reference	91
12.15 MetNoFimex::CDMDimension Class Reference	95
12.16 MetNoFimex::CDMException Class Reference	97
12.17 MetNoFimex::CDMExtractor Class Reference	98
12.18 MetNoFimex::CDMInterpolator Class Reference	101
12.19 MetNoFimex::CDMNameCompare Struct Reference	104
12.20 MetNoFimex::CDMNamedEntity Class Reference	105
12.21 MetNoFimex::CDMNameEqual Class Reference	106
12.22 MetNoFimex::CDMReader Class Reference	107
12.23 MetNoFimex::CDMTimeInterpolator Class Reference	110
12.24 MetNoFimex::CDMVariable Class Reference	112
12.25 MetNoFimex::CDMWriter Class Reference	115
12.26 MetNoFimex::Data Class Reference	116
12.27 MetNoFimex::DataImpl< C > Class Template Reference	121
12.28 MetNoFimex::DataTypeChanger Class Reference	129
12.29 MetNoFelt::Felt_Array Class Reference	131
12.30 MetNoFelt::Felt_File Class Reference	136
12.31 MetNoFelt::Felt_File_Error Class Reference	140
12.32 MetNoFimex::FeltCDMReader Class Reference	141
12.33 MetNoFelt::FeltParameters Class Reference	142
12.34 MetNoFimex::FimexTime Class Reference	144

12.35	MetNoFimex::GribApiCDMWriter Class Reference	147
12.36	MetNoFimex::GribApiCDMWriter_Impl1 Class Reference	148
12.37	MetNoFimex::GribApiCDMWriter_Impl2 Class Reference	150
12.38	MetNoFimex::GribApiCDMWriter_ImplAbstract Class Reference	152
12.39	KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc > Class Template Reference	156
12.40	MetNoFimex::Logger Class Reference	187
12.41	MetNoFimex::NcmlCDMReader Class Reference	189
12.42	MetNoFimex::NetCDF_CDMWriter Class Reference	191
12.43	MetNoFimex::NetCDF_CF10_CDMReader Class Reference	193
12.44	MetNoFimex::Null_CDMWriter Class Reference	195
12.45	MetNoFimex::ReplaceStringObject Class Reference	196
12.46	MetNoFimex::ReplaceStringTimeObject Class Reference	198
12.47	MetNoFimex::ShortPairLess Struct Reference	200
12.48	MetNoFimex::SpatialAxisSpec Class Reference	201
12.49	KDTree::squared_difference< _Tp, _Dist > Struct Template Reference	203
12.50	KDTree::squared_difference_counted< _Tp, _Dist > Struct Template Reference	204
12.51	MetNoFimex::TimeLevelDataSliceFetcher Class Reference	205
12.52	MetNoFimex::TimeSpec Class Reference	206
12.53	MetNoFimex::TimeUnit Class Reference	208
12.54	MetNoFimex::UnitException Class Reference	210
12.55	MetNoFimex::Units Class Reference	211
12.56	MetNoFimex::XMLDoc Class Reference	213
13	File Documentation	215
13.1	doxydoc.txt File Reference	215
13.2	include/fimex/CachedInterpolation.h File Reference	216
13.3	include/fimex/CachedVectorReprojection.h File Reference	217
13.4	include/fimex/CDM.h File Reference	218
13.5	include/fimex/CDMAttribute.h File Reference	219
13.6	include/fimex/CDMconstants.h File Reference	220
13.7	include/fimex/CDMDataType.h File Reference	221
13.8	include/fimex/CDMDimension.h File Reference	222
13.9	include/fimex/CDMException.h File Reference	223
13.10	include/fimex/CDMExtractor.h File Reference	224
13.11	include/fimex/CDMInterpolator.h File Reference	225
13.12	include/fimex/CDMNamedEntity.h File Reference	226
13.13	include/fimex/CDMReader.h File Reference	227

13.14include/fimex/CDMTimeInterpolator.h File Reference	228
13.15include/fimex/CDMVariable.h File Reference	229
13.16include/fimex/CDMWriter.h File Reference	230
13.17include/fimex/config.h File Reference	231
13.18include/fimex/Data.h File Reference	234
13.19include/fimex/DataImpl.h File Reference	235
13.20include/fimex/DataTypeChanger.h File Reference	236
13.21include/fimex/Felt_Array.h File Reference	237
13.22include/fimex/Felt_File.h File Reference	238
13.23include/fimex/Felt_File_Error.h File Reference	239
13.24include/fimex/FeltCDMReader.h File Reference	240
13.25include/fimex/FeltParameters.h File Reference	241
13.26include/fimex/GribApiCDMWriter.h File Reference	242
13.27include/fimex/GribApiCDMWriter_Impl1.h File Reference	243
13.28include/fimex/GribApiCDMWriter_Impl2.h File Reference	244
13.29include/fimex/GribApiCDMWriter_ImplAbstract.h File Reference	245
13.30include/fimex/interpolation.h File Reference	246
13.31include/fimex/Logger.h File Reference	256
13.32include/fimex/NcmlCDMReader.h File Reference	257
13.33include/fimex/NetCDF_CDMWriter.h File Reference	258
13.34include/fimex/NetCDF_CF10_CDMReader.h File Reference	259
13.35include/fimex/NetCDF_Utils.h File Reference	260
13.36include/fimex/Null_CDMWriter.h File Reference	261
13.37include/fimex/ReplaceStringObject.h File Reference	262
13.38include/fimex/ReplaceStringTimeObject.h File Reference	263
13.39include/fimex/SpatialAxisSpec.h File Reference	264
13.40include/fimex/TimeLevelDataSliceFetcher.h File Reference	265
13.41include/fimex/TimeSpec.h File Reference	266
13.42include/fimex/TimeUnit.h File Reference	267
13.43include/fimex/Units.h File Reference	268
13.44include/fimex/Utils.h File Reference	269
13.45include/fimex/XMLDoc.h File Reference	270
13.46include/kdtree++/allocator.hpp File Reference	271
13.47include/kdtree++/function.hpp File Reference	272
13.48include/kdtree++/iterator.hpp File Reference	273
13.49include/kdtree++/kdtree.hpp File Reference	274

13.50include/kdtree++/node.hpp File Reference	276
13.51include/kdtree++/region.hpp File Reference	277

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](#)
- [ncmlConfugration](#)
- [gribWriter Configuration](#)
- [netcdfWriter 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
--output.file arg         output file
--output.type arg         filetype of output file
--output.config arg       non-standard output configuration
--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
--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 x-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 uunits string,
                           i.e. m or degrees_east
--interpolate.yAxisUnit arg unit of y-Axis given as uunits 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
--timeInterpolate.timeSpec arg specification of times to interpolate to,
                           see Fimex::TimeSpec for a full definition
--timeInterpolate.printNcML print NcML description of timeInterpolator
--ncml.config              modify/configure with ncml-file
--ncml.printNcML          print NcML description after ncml-configuration
```

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

```
# config file for program fimex
[input]
file=flth00.dat
```

```
config=../share/etc/felt2nc_variables.xml
type=felt

[output]
file=test.nc
type=netcdf

[extract]
removeVariable=relative_humidity

[extract.reduceDimension]
name=x
start=2
end=-2

[extract.reduceDimension]
name=y
start=2
end=-2

[interpolate]
method = bilinear
projString = +proj=stere +lat_0=90 +lon_0=-32 +lat_ts=60 +elips=sphere +a=6371000 +e=0
xAxisValues = -500000,-450000,...,500000
yAxisValues = -500000,-450000,...,500000
xAxisUnit = m
yAxisUnit = m

[timeInterpolate]
timeSpec = 0,3,...,x;relativeUnit=hours since 2001-01-01 10:00:00;unit=hours since 2007-05-16 00:00:00
```

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 [MetNoFimex::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.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE cdm_felt_config SYSTEM "felt2nc_variables.dtd">
<cdm_felt_config>
<!-- optional processing options for felt-files -->
<processOptions>
  <!-- allowed deviation of gridParameter (6 values, see felt.doc) to still assumed to be the same grid,
  <!-- <option name="gridParameterDelta" value="0 0 0 0 0 0"/> -->
</processOptions>
<global_attributes>
<attribute name="Conventions" value="CF-1.0" type="string" />
<attribute name="institution" value="Norwegian Meteorological Institute, met.no" type="string" />
<attribute name="source" value="HIRLAM" type="string" />
<attribute name="title" value="unknown" type="string" />
<attribute name="min_time" value="%MIN_DATETIME(%Y-%m-%d %H:%M:%SZ)%" type="string" />
<attribute name="max_time" value="%MAX_DATETIME(%Y-%m-%d)%" type="string" />
```



```
<attribute name="Expires" value="%MAX_DATETIME(%Y-%m-%d,2419200)%" type="string" />
<attribute name="references" value="unknown" type="string" />
<attribute name="history" value="unknown" type="string" />
<attribute name="comment" value="none" type="string" />
</global_attributes>
<axes>
<time id="time" name="time" type="double">
<attribute name="long_name" value="time" type="string" />
<attribute name="standard_name" value="time" type="string" />
<attribute name="units" value="seconds since 1970-01-01 00:00:00 +00:00" type="string" />
</time>
<!-- polar-stereographic at 60deg -->
<spatial_axis projection_felt_id="1" id="x" name="x" type="int">
<attribute name="long_name" value="x-coordinate in Cartesian system" type="string" />
<attribute name="standard_name" value="projection_x_coordinate" type="string" />
<attribute name="units" value="m" type="string" />
</spatial_axis>
<spatial_axis projection_felt_id="1" id="y" name="y" type="int">
<attribute name="long_name" value="y-coordinate in Cartesian system" type="string" />
<attribute name="standard_name" value="projection_y_coordinate" type="string" />
<attribute name="units" value="m" type="string" />
</spatial_axis>
<!-- geographic -->
<spatial_axis projection_felt_id="2" id="x" name="lon" type="float">
<attribute name="long_name" value="longitude" type="string" />
<attribute name="units" value="degrees_east" type="string" />
</spatial_axis>
<spatial_axis projection_felt_id="2" id="y" name="lat" type="float">
<attribute name="long_name" value="latitude" type="string" />
<attribute name="units" value="degrees_north" type="string" />
</spatial_axis>
<!-- spherical rotated -->
<spatial_axis projection_felt_id="3" id="x" name="r lon" type="float">
<attribute name="long_name" value="rotated longitude" type="string" />
<attribute name="standard_name" value="grid_longitude" type="string" />
<attribute name="units" value="degrees" type="string" />
</spatial_axis>
<spatial_axis projection_felt_id="3" id="y" name="r lat" type="float">
<attribute name="long_name" value="rotated latitude" type="string" />
<attribute name="standard_name" value="grid_latitude" type="string" />
<attribute name="units" value="degrees" type="string" />
</spatial_axis>
<!-- polar-stereographic -->
<spatial_axis projection_felt_id="4" id="x" name="x" type="int">
<attribute name="long_name" value="x-coordinate in Cartesian system" type="string" />
<attribute name="standard_name" value="projection_x_coordinate" type="string" />
<attribute name="units" value="m" type="string" />
</spatial_axis>
<spatial_axis projection_felt_id="4" id="y" name="y" type="int">
<attribute name="long_name" value="y-coordinate in Cartesian system" type="string" />
<attribute name="standard_name" value="projection_y_coordinate" type="string" />
<attribute name="units" value="m" type="string" />
</spatial_axis>
<!-- mercator -->
<spatial_axis projection_felt_id="5" id="x" name="x" type="int">
<attribute name="long_name" value="x-coordinate in Cartesian system" type="string" />
<attribute name="standard_name" value="projection_x_coordinate" type="string" />
<attribute name="units" value="m" type="string" />
</spatial_axis>
<spatial_axis projection_felt_id="5" id="y" name="y">
<attribute name="long_name" value="y-coordinate in Cartesian system" type="string" />
<attribute name="standard_name" value="projection_y_coordinate" type="string" />
<attribute name="units" value="m" type="string" />
</spatial_axis>
<spatial_axis id="longitude" name="longitude">
<attribute name="valid_max" value="180." type="float" />
<attribute name="valid_min" value="-180." type="float" />
```

```

<attribute name="long_name" value="longitude" type="string" />
<attribute name="standard_name" value="longitude" type="string" />
<attribute name="units" value="degree_east" type="string" />
</spatial_axis>
<spatial_axis id="latitude" name="latitude">
<attribute name="valid_max" value="90." type="float" />
<attribute name="valid_min" value="-90." type="float" />
<attribute name="long_name" value="latitude" type="string" />
<attribute name="standard_name" value="latitude" type="string" />
<attribute name="units" value="degree_north" type="string" />
</spatial_axis>
<vertical_axis id="pressure" name="pressure" felt_id="1" type="short">
<attribute name="description" value="pressure" type="string" />
<attribute name="long_name" value="pressure" type="string" />
<attribute name="standard_name" value="air_pressure" type="string" />
<attribute name="positive" value="up" type="string" />
<attribute name="units" value="hPa" type="string" />
</vertical_axis>
<vertical_axis id="sigma" name="sigma" felt_id="2" type="short">
<attribute name="description" value="atmosphere sigma coordinate" type="string" />
<attribute name="long_name" value="atmosphere_sigma_coordinate" type="string" />
<attribute name="standard_name" value="atmosphere_sigma_coordinate" type="string" />
<attribute name="positive" value="up" type="string" />
<attribute name="scale_factor" value="0.001" type="float" />
</vertical_axis>
<vertical_axis id="h" name="h" felt_id="3" type="short">
<attribute name="description" value="vertical coordinate used for wave variables, value: 0" type="string" />
<attribute name="long_name" value="vertical_wave_coordinate" type="string" />
</vertical_axis>
<vertical_axis id="theta" name="theta" felt_id="4" type="short">
<attribute name="description" value="isentrop layer?" type="string" />
</vertical_axis>
<vertical_axis id="depth" name="depth" felt_id="5" type="short">
<attribute name="description" value="geopotential level relative to equilibrium surface" type="string" />
<attribute name="long_name" value="depth" type="string" />
<attribute name="positive" value="down" type="string" />
<attribute name="standard_name" value="depth" type="string" />
<attribute name="units" value="m" type="string" />
</vertical_axis>
<vertical_axis id="layer" name="layer" felt_id="6" type="short">
<attribute name="description" value="ocean model layer no., isopycnic or hybrid" type="string" />
<attribute name="long_name" value="ocean_layer_coordinate" type="string" />
<attribute name="positive" value="down" type="string" />
</vertical_axis>
<vertical_axis id="sigma" name="sigma" felt_id="7" type="short">
<attribute name="description" value="ocean sigma coordinate, surface is 0, bottom is 1" type="string" />
<attribute name="long_name" value="ocean_sigma_coordinate" type="string" />
<attribute name="positive" value="down" type="string" />
<attribute name="standard_name" value="ocean_sigma_coordinate" type="string" />
<attribute name="scale_factor" value="0.001" type="float" />
</vertical_axis>
<vertical_axis id="surface" name="surface" felt_id="8" type="short">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="description" value="ocean surface, or vertically integrated" type="string" />
<attribute name="long_name" value="model_surface" type="string" />
<attribute name="positive" value="up" type="string" />
</vertical_axis>
<vertical_axis id="k" name="k" felt_id="10" type="double">
<attribute name="standard_name" value="atmosphere_hybrid_sigma_pressure_coordinate" type="string" />
<attribute name="formular" value="p(n,k,j,i) = ap(k) + b(k)*ps(n,j,i)" type="string" />
<attribute name="formula_terms" value="ap: ap b: b ps: ps p0: p0" type="string" />
<attribute name="long_name" value="atmosphere_hybrid_sigma_pressure_coordinate" type="string" />
<attribute name="positive" value="up" type="string" />
<values mode="hybridSigmaCalc(ap,b)" />
<!-- optional values, will otherwise be calculated -->
<!-- <values mode="inline">0.01000025677 0.030167302165 0.0506574118 0.071450009935 0.09252333188 0.113854
<additional_axis_variable name="ap" type="double" axis="k">

```

```

<attribute name="units" value="Pa" type="string" />
<values mode="level2" scale_factor="10" />
<!-- optional values, will otherwise be retrieved from level2 -->
<!-- <values>1000.025677 3016.7302165 5053.90618 7087.019935 9093.765188 11053.98013 12949.566675 14764.
</additional_axis_variable>
<additional_axis_variable name="b" type="double" axis="k">
<attribute name="units" value="1" type="string" />
<values mode="hybridLevels" scale_factor="0.0001" />
<!-- optional values, will otherwise be retrieved from ident19 -->
<!-- <values>0 0 0.00011835 0.00057981 0.00158568 0.003314615 0.00592347 0.00954814 0.01430438 0.02028863
</additional_axis_variable>
<!-- currently not possible to read axis independent variable
<additional_axis_variable name="p0" type="int" axis="1">
<attribute name="long_name" value="reference pressure for hybrid sigma coordinate" type="string" />
<attribute name="units" value="Pa" type="string" />
<values>100000</values>
</additional_axis_variable>
-->
</vertical_axis>
</axes>
<variables>
<!--
<parameter id="1" name="geopotential_height">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="long_name" value="geopotential_height" type="string" />
<attribute name="standard_name" value="geopotential_height" type="string" />
<attribute name="units" value="m" type="string" />
</parameter>
-->
<!--
<parameter id="2" name="x_wind">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="long_name" value="x_wind" type="string" />
<attribute name="standard_name" value="x_wind" type="string" />
<attribute name="units" value="m/s" type="string" />
<spatial_vector direction="x,longitude" counterpart="y_wind" />
</parameter>
<parameter id="3" name="y_wind">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="long_name" value="y_wind" type="string" />
<attribute name="standard_name" value="y_wind" type="string" />
<attribute name="units" value="m/s" type="string" />
<spatial_vector direction="y,latitude" counterpart="x_wind" />
</parameter>
-->
<!--
<parameter id="10" name="relative_humidity">
<attribute name="scale_factor" value="0.01" type="float" />
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="long_name" value="relative_humidity" type="string" />
<attribute name="standard_name" value="relative_humidity" type="string" />
<attribute name="units" value="1" type="string" />
</parameter>
-->
<parameter id="17,2,1000" name="precipitation_amount" type="float">
<attribute name="_FillValue" value="-32767" type="float" />
<attribute name="long_name" value="precipitation_amount" type="string" />
<attribute name="standard_name" value="precipitation_amount" type="string" />
<attribute name="units" value="kg/m2" type="string" />
</parameter>
<!--
<parameter id="18" name="air_potential_temperature">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="long_name" value="air_potential_temperature" type="string" />
<attribute name="standard_name" value="air_potential_temperature" type="string" />
<attribute name="units" value="K" type="string" />
</parameter>

```

```

-->
<parameter id="25,2,1000" name="cloud_area_fraction">
<attribute name="scale_factor" value="0.01" type="float" />
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="cloud_area_fraction" type="string" />
<attribute name="standard_name" value="cloud_area_fraction" type="string" />
<attribute name="units" value="1" type="string" />
</parameter>
<parameter id="31,2,1000" name="air_temperature">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point height: p" type="string" />
<attribute name="long_name" value="air_temperature" type="string" />
<attribute name="standard_name" value="air_temperature" type="string" />
<attribute name="units" value="K" type="string" />
</parameter>
<parameter id="32,2,1000" name="relative_humidity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="scale_factor" value="0.01" type="float" />
<attribute name="cell_methods" value="time: point height: 2m" type="string" />
<attribute name="long_name" value="relative_humidity" type="string" />
<attribute name="standard_name" value="relative_humidity" type="string" />
<attribute name="units" value="1" type="string" />
</parameter>
<parameter id="33,2,1000" name="x_wind_10m" type="float">
<attribute name="_FillValue" value="9.9692099683868690e+36f" type="float" />
<attribute name="cell_methods" value="time: point height: 10m" type="string" />
<attribute name="long_name" value="x_wind_10m" type="string" />
<attribute name="metno_name" value="x_wind_10m" type="string" />
<attribute name="units" value="m s-1" type="string" />
<spatial_vector direction="x,longitude" counterpart="y_wind_10m" />
</parameter>
<parameter id="34,2,1000" name="y_wind_10m" type="float">
<attribute name="_FillValue" value="9.9692099683868690e+36f" type="float" />
<attribute name="cell_methods" value="time: point height: 10m" type="string" />
<attribute name="long_name" value="y_wind_10m" type="string" />
<attribute name="standard_name" value="y_wind_10m" type="string" />
<attribute name="units" value="m s-1" type="string" />
<spatial_vector direction="y,latitude" counterpart="x_wind_10m" />
</parameter>
<!-- change in scale_factor
<parameter id="36,2,1000" name="accumulated_surface_upward_sensible_heat_flux">
<attribute name="long_name" value="accumulated_surface_upward_sensible_heat_flux" type="string" />
<attribute name="metno_name" value="accumulated_surface_upward_sensible_heat_flux" type="string" />
<attribute name="units" value="kJ/m2" type="string" />
</parameter>
<parameter id="37,2,1000" name="accumulated_surface_upward_latent_heat_flux">
<attribute name="long_name" value="accumulated_surface_upward_latent_heat_flux" type="string" />
<attribute name="metno_name" value="accumulated_surface_upward_latent_heat_flux" type="string" />
<attribute name="units" value="kJ/m2" type="string" />
</parameter>
-->
<parameter id="39" name="cloud_area_fraction_in_atmosphere_layer">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="long_name" value="cloud_area_fraction_in_atmosphere_layer" type="string" />
<attribute name="standard_name" value="cloud_area_fraction_in_atmosphere_layer" type="string" />
<attribute name="scale_factor" value="0.01" type="float" />
<attribute name="units" value="1" type="string" />
</parameter>
<parameter id="58,2,1000" name="sea_level_pressure">
<attribute name="scale_factor" value="100." type="float" />
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="air_pressure_at_sea_level" type="string" />
<attribute name="standard_name" value="air_pressure_at_sea_level" type="string" />
<attribute name="units" value="Pa" type="string" />
</parameter>

```

```
<parameter id="66,2,1000" name="surface_snow_sickness">
<attribute name="long_name" value="surface_snow_sickness" type="string" />
<attribute name="standard_name" value="surface_snow_sickness" type="string" />
<attribute name="units" value="m" type="string" />
</parameter>
<parameter id="101,2,1000" name="altitude" type="short">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="long_name" value="altitude" type="string" />
<attribute name="standard_name" value="altitude" type="string" />
<attribute name="units" value="m" type="string" />
</parameter>
<parameter id="103,2,1000" name="sea_surface_temperature">
<attribute name="long_name" value="sea_surface_temperature" type="string" />
<attribute name="metno_name" value="sea_surface_temperature" type="string" />
<attribute name="units" value="K" type="string" />
</parameter>
<parameter id="191,2,1000" name="land_ice_area_fraction">
<attribute name="long_name" value="land_ice_area_fraction" type="string" />
<attribute name="metno_name" value="land_ice_area_fraction" type="string" />
<attribute name="scale_factor" value="0.01" type="float" />
<attribute name="units" value="1" type="string" />
</parameter>
<parameter id="200,3,0" name="significant_wave_height">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_surface_wave_significant_height" type="string" />
<attribute name="standard_name" value="sea_surface_wave_significant_height" type="string" />
<attribute name="units" value="m" type="string" />
</parameter>
<parameter id="201,3,0" name="peak_wave_period">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_surface_wave_peak_period" type="string" />
<attribute name="units" value="s" type="string" />
</parameter>
<parameter id="202,3,0" name="significant_wave_period">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_surface_wave_significant_period" type="string" />
<attribute name="units" value="s" type="string" />
</parameter>
<parameter id="203,3,0" name="peak_wave_direction">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_surface_wave_peak_to_direction" type="string" />
<attribute name="units" value="degree" type="string" />
</parameter>
<parameter id="204,3,0" name="wave_direction">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_surface_wave_to_direction" type="string" />
<attribute name="standard_name" value="sea_surface_wave_to_direction" type="string" />
<attribute name="units" value="degree" type="string" />
</parameter>
<parameter id="210,3,0" name="significant_wind_wave_height">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_surface_wind_wave_significant_height" type="string" />
<attribute name="standard_name" value="sea_surface_wind_wave_significant_height" type="string" />
<attribute name="units" value="m" type="string" />
</parameter>
<parameter id="213,3,0" name="sea_surface_wind_wave_period">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_surface_wind_wave_period" type="string" />
<attribute name="standard_name" value="sea_surface_wind_wave_period" type="string" />
<attribute name="units" value="s" type="string" />
</parameter>
```

```

</parameter>
<parameter id="214,3,0" name="sea_surface_wind_wave_to_direction">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_surface_wind_wave_to_direction" type="string" />
<attribute name="standard_name" value="sea_surface_wind_wave_to_direction" type="string" />
<attribute name="units" value="degree" type="string" />
</parameter>
<parameter id="220,3,0" name="significant_swell_wave_height">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_surface_swell_wave_significant_height" type="string" />
<attribute name="standard_name" value="sea_surface_swell_wave_significant_height" type="string" />
<attribute name="units" value="m" type="string" />
</parameter>
<parameter id="223,3,0" name="sea_surface_swell_wave_period">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_surface_swell_wave_period" type="string" />
<attribute name="standard_name" value="sea_surface_swell_wave_period" type="string" />
<attribute name="units" value="s" type="string" />
</parameter>
<parameter id="224,3,0" name="sea_surface_swell_wave_to_direction">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_surface_swell_wave_to_direction" type="string" />
<attribute name="standard_name" value="sea_surface_swell_wave_to_direction" type="string" />
<attribute name="units" value="degree" type="string" />
</parameter>
<parameter id="291,3,0" name="x_stokes_drift">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="x_sea_wave_stokes_drift" type="string" />
<attribute name="units" value="m s-1" type="string" />
</parameter>
<parameter id="292,3,0" name="y_stokes_drift">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="y_sea_wave_stokes_drift" type="string" />
<attribute name="units" value="m s-1" type="string" />
</parameter>
<parameter id="301,8,0" name="sea_surface_height">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_surface_height_above_geoid" type="string" />
<attribute name="standard_name" value="sea_surface_height_above_geoid" type="string" />
<attribute name="units" value="m" type="string" />
</parameter>
<parameter id="302,5,0" name="x_velocity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_water_x_velocity" type="string" />
<attribute name="standard_name" value="sea_water_x_velocity" type="string" />
<attribute name="units" value="m s-1" type="string" />
</parameter>
<parameter id="302,6,0" name="x_velocity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_water_x_velocity" type="string" />
<attribute name="standard_name" value="sea_water_x_velocity" type="string" />
<attribute name="units" value="m s-1" type="string" />
</parameter>
<parameter id="302,7,0" name="x_velocity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_water_x_velocity" type="string" />
<attribute name="standard_name" value="sea_water_x_velocity" type="string" />

```

```
<attribute name="units" value="m s-1" type="string" />
</parameter>
<parameter id="302,8,0" name="x_btrop_velocity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="barotropic_sea_water_x_velocity" type="string" />
<attribute name="standard_name" value="sea_water_x_velocity" type="string" />
<attribute name="units" value="m s-1" type="string" />
</parameter>
<parameter id="303,5,0" name="y_velocity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_water_y_velocity" type="string" />
<attribute name="standard_name" value="sea_water_y_velocity" type="string" />
<attribute name="units" value="m s-1" type="string" />
</parameter>
<parameter id="303,6,0" name="y_velocity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_water_y_velocity" type="string" />
<attribute name="standard_name" value="sea_water_y_velocity" type="string" />
<attribute name="units" value="m s-1" type="string" />
</parameter>
<parameter id="303,7,0" name="y_velocity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_water_y_velocity" type="string" />
<attribute name="standard_name" value="sea_water_y_velocity" type="string" />
<attribute name="units" value="m s-1" type="string" />
</parameter>
<parameter id="303,8,0" name="y_btrop_velocity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="barotropic_sea_water_y_velocity" type="string" />
<attribute name="standard_name" value="sea_water_y_velocity" type="string" />
<attribute name="units" value="m s-1" type="string" />
</parameter>
<parameter id="304,5,0" name="z_velocity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="upward_sea_water_velocity" type="string" />
<attribute name="standard_name" value="upward_sea_water_velocity" type="string" />
<attribute name="units" value="m s-1" type="string" />
</parameter>
<parameter id="304,7,0" name="z_velocity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="upward_sea_water_velocity" type="string" />
<attribute name="standard_name" value="upward_sea_water_velocity" type="string" />
<attribute name="units" value="m s-1" type="string" />
</parameter>
<parameter id="305,5,0" name="sigma_t">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_water_sigma_t" type="string" />
<attribute name="standard_name" value="sea_water_sigma_t" type="string" />
<attribute name="units" value="kg m-3" type="string" />
</parameter>
<parameter id="305,6,0" name="sigma_t">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_water_sigma_t" type="string" />
<attribute name="standard_name" value="sea_water_sigma_t" type="string" />
<attribute name="units" value="kg m-3" type="string" />
</parameter>
<parameter id="305,7,0" name="sigma_t">
<attribute name="_FillValue" value="-32767" type="short" />
```

```

<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_water_sigma_t" type="string" />
<attribute name="standard_name" value="sea_water_sigma_t" type="string" />
<attribute name="units" value="kg m-3" type="string" />
</parameter>
<parameter id="306,6,0" name="layer_thickness">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="valid_min" value="0" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_layer_thickness" type="string" />
<attribute name="units" value="m" type="string" />
</parameter>
<parameter id="306,7,0" name="layer_thickness">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="valid_min" value="0" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_layer_thickness" type="string" />
<attribute name="units" value="m" type="string" />
</parameter>
<parameter id="307,5,0" name="salinity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="valid_min" value="0" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_water_salinity" type="string" />
<attribute name="standard_name" value="sea_water_salinity" type="string" />
<attribute name="units" value="1e-3" type="string" />
</parameter>
<parameter id="307,6,0" name="salinity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="valid_min" value="0" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_water_salinity" type="string" />
<attribute name="standard_name" value="sea_water_salinity" type="string" />
<attribute name="units" value="1e-3" type="string" />
</parameter>
<parameter id="307,7,0" name="salinity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="valid_min" value="0" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_water_salinity" type="string" />
<attribute name="standard_name" value="sea_water_salinity" type="string" />
<attribute name="units" value="1e-3" type="string" />
</parameter>
<parameter id="307,8,0" name="salinity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="valid_min" value="0" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_surface_salinity" type="string" />
<attribute name="standard_name" value="sea_surface_salinity" type="string" />
<attribute name="units" value="1e-3" type="string" />
</parameter>
<parameter id="308,5,0" name="sea_temperature">
<attribute name="add_offset" value="273.15" type="float" />
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_water_temperature" type="string" />
<attribute name="standard_name" value="sea_water_temperature" type="string" />
<attribute name="units" value="K" type="string" />
</parameter>
<parameter id="308,6,0" name="sea_temperature">
<attribute name="add_offset" value="273.15" type="float" />
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_water_temperature" type="string" />
<attribute name="standard_name" value="sea_water_temperature" type="string" />
<attribute name="units" value="K" type="string" />
</parameter>

```



```
<parameter id="308,7,0" name="sea_temperature">
<attribute name="add_offset" value="273.15" type="float" />
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_water_temperature" type="string" />
<attribute name="standard_name" value="sea_water_temperature" type="string" />
<attribute name="units" value="K" type="string" />
</parameter>
<parameter id="308,8,0" name="sea_surface_temperature">
<attribute name="add_offset" value="273.15" type="float" />
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_surface_temperature" type="string" />
<attribute name="standard_name" value="sea_surface_temperature" type="string" />
<attribute name="units" value="K" type="string" />
</parameter>
<parameter id="312,5,0" name="TKE">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="valid_min" value="0" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_water_turbulent_kinetic_energy" type="string" />
<attribute name="units" value="m2 s-2" type="string" />
</parameter>
<parameter id="312,6,0" name="TKE">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="valid_min" value="0" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_water_turbulent_kinetic_energy" type="string" />
<attribute name="units" value="m2 s-2" type="string" />
</parameter>
<parameter id="312,7,0" name="TKE">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="valid_min" value="0" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_water_turbulent_kinetic_energy" type="string" />
<attribute name="units" value="m2 s-2" type="string" />
</parameter>
<parameter id="331,8,0" name="sea_surface_height">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: mean" type="string" />
<attribute name="long_name" value="sea_surface_height_above_geoid" type="string" />
<attribute name="standard_name" value="sea_surface_height_above_geoid" type="string" />
<attribute name="units" value="m" type="string" />
</parameter>
<parameter id="332,5,0" name="x_velocity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: mean" type="string" />
<attribute name="long_name" value="sea_water_x_velocity" type="string" />
<attribute name="standard_name" value="sea_water_x_velocity" type="string" />
<attribute name="units" value="m s-1" type="string" />
</parameter>
<parameter id="332,6,0" name="x_velocity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: mean" type="string" />
<attribute name="long_name" value="sea_water_x_velocity" type="string" />
<attribute name="standard_name" value="sea_water_x_velocity" type="string" />
<attribute name="units" value="m s-1" type="string" />
</parameter>
<parameter id="332,7,0" name="x_velocity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: mean" type="string" />
<attribute name="long_name" value="sea_water_x_velocity" type="string" />
<attribute name="standard_name" value="sea_water_x_velocity" type="string" />
<attribute name="units" value="m s-1" type="string" />
</parameter>
<parameter id="332,8,0" name="x_btrop_velocity">
<attribute name="_FillValue" value="-32767" type="short" />
```

```

<attribute name="cell_methods" value="time: mean" type="string" />
<attribute name="long_name" value="barotropic_sea_water_x_velocity" type="string" />
<attribute name="standard_name" value="sea_water_x_velocity" type="string" />
<attribute name="units" value="m s-1" type="string" />
</parameter>
<parameter id="333,5,0" name="y_velocity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: mean" type="string" />
<attribute name="long_name" value="sea_water_y_velocity" type="string" />
<attribute name="standard_name" value="sea_water_y_velocity" type="string" />
<attribute name="units" value="m s-1" type="string" />
</parameter>
<parameter id="333,6,0" name="y_velocity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: mean" type="string" />
<attribute name="long_name" value="sea_water_y_velocity" type="string" />
<attribute name="standard_name" value="sea_water_y_velocity" type="string" />
<attribute name="units" value="m s-1" type="string" />
</parameter>
<parameter id="333,7,0" name="y_velocity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: mean" type="string" />
<attribute name="long_name" value="sea_water_y_velocity" type="string" />
<attribute name="standard_name" value="sea_water_y_velocity" type="string" />
<attribute name="units" value="m s-1" type="string" />
</parameter>
<parameter id="333,8,0" name="y_btrop_velocity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: mean" type="string" />
<attribute name="long_name" value="barotropic_sea_water_y_velocity" type="string" />
<attribute name="standard_name" value="sea_water_y_velocity" type="string" />
<attribute name="units" value="m s-1" type="string" />
</parameter>
<parameter id="334,5,0" name="z_velocity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: mean" type="string" />
<attribute name="long_name" value="upward_sea_water_velocity" type="string" />
<attribute name="standard_name" value="upward_sea_water_velocity" type="string" />
<attribute name="units" value="m s-1" type="string" />
</parameter>
<parameter id="334,7,0" name="z_velocity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: mean" type="string" />
<attribute name="long_name" value="upward_sea_water_velocity" type="string" />
<attribute name="standard_name" value="upward_sea_water_velocity" type="string" />
<attribute name="units" value="m s-1" type="string" />
</parameter>
<parameter id="335,5,0" name="sigma_t">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: mean" type="string" />
<attribute name="long_name" value="sea_water_sigma_t" type="string" />
<attribute name="standard_name" value="sea_water_sigma_t" type="string" />
<attribute name="units" value="kg m-3" type="string" />
</parameter>
<parameter id="335,6,0" name="sigma_t">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: mean" type="string" />
<attribute name="long_name" value="sea_water_sigma_t" type="string" />
<attribute name="standard_name" value="sea_water_sigma_t" type="string" />
<attribute name="units" value="kg m-3" type="string" />
</parameter>
<parameter id="335,7,0" name="sigma_t">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: mean" type="string" />
<attribute name="long_name" value="sea_water_sigma_t" type="string" />
<attribute name="standard_name" value="sea_water_sigma_t" type="string" />
<attribute name="units" value="kg m-3" type="string" />

```

```
</parameter>
<parameter id="336,6,0" name="layer_thickness">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="valid_min" value="0" type="short" />
<attribute name="cell_methods" value="time: mean" type="string" />
<attribute name="long_name" value="sea_layer_thickness" type="string" />
<attribute name="units" value="m" type="string" />
</parameter>
<parameter id="336,7,0" name="layer_thickness">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="valid_min" value="0" type="short" />
<attribute name="cell_methods" value="time: mean" type="string" />
<attribute name="long_name" value="sea_layer_thickness" type="string" />
<attribute name="units" value="m" type="string" />
</parameter>
<parameter id="337,5,0" name="salinity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="valid_min" value="0" type="short" />
<attribute name="cell_methods" value="time: mean" type="string" />
<attribute name="long_name" value="sea_water_salinity" type="string" />
<attribute name="standard_name" value="sea_water_salinity" type="string" />
<attribute name="units" value="1e-3" type="string" />
</parameter>
<parameter id="337,6,0" name="salinity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="valid_min" value="0" type="short" />
<attribute name="cell_methods" value="time: mean" type="string" />
<attribute name="long_name" value="sea_water_salinity" type="string" />
<attribute name="standard_name" value="sea_water_salinity" type="string" />
<attribute name="units" value="1e-3" type="string" />
</parameter>
<parameter id="337,7,0" name="salinity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="valid_min" value="0" type="short" />
<attribute name="cell_methods" value="time: mean" type="string" />
<attribute name="long_name" value="sea_water_salinity" type="string" />
<attribute name="standard_name" value="sea_water_salinity" type="string" />
<attribute name="units" value="1e-3" type="string" />
</parameter>
<parameter id="337,8,0" name="salinity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="valid_min" value="0" type="short" />
<attribute name="cell_methods" value="time: mean" type="string" />
<attribute name="long_name" value="sea_surface_salinity" type="string" />
<attribute name="standard_name" value="sea_surface_salinity" type="string" />
<attribute name="units" value="1e-3" type="string" />
</parameter>
<parameter id="338,5,0" name="temperature">
<attribute name="add_offset" value="273.15" type="float" />
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: mean" type="string" />
<attribute name="long_name" value="sea_water_temperature" type="string" />
<attribute name="standard_name" value="sea_water_temperature" type="string" />
<attribute name="units" value="K" type="string" />
</parameter>
<parameter id="338,6,0" name="temperature">
<attribute name="add_offset" value="273.15" type="float" />
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: mean" type="string" />
<attribute name="long_name" value="sea_water_temperature" type="string" />
<attribute name="standard_name" value="sea_water_temperature" type="string" />
<attribute name="units" value="K" type="string" />
</parameter>
<parameter id="338,7,0" name="temperature">
<attribute name="add_offset" value="273.15" type="float" />
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: mean" type="string" />
```

```

<attribute name="long_name" value="sea_water_temperature" type="string" />
<attribute name="standard_name" value="sea_water_temperature" type="string" />
<attribute name="units" value="K" type="string" />
</parameter>
<parameter id="338,8,0" name="temperature">
<attribute name="add_offset" value="273.15" type="float" />
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: mean" type="string" />
<attribute name="long_name" value="sea_surface_temperature" type="string" />
<attribute name="standard_name" value="sea_surface_temperature" type="string" />
<attribute name="units" value="K" type="string" />
</parameter>
<parameter id="339,2,0" name="total_precipitation_rate">
<attribute name="valid_min" value="0" type="short" />
<attribute name="scale_factor" value="4.630e-8" type="float" />
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="lwe_precipitation_rate" type="string" />
<attribute name="standard_name" value="lwe_precipitation_rate" type="string" />
<attribute name="units" value="m s-1" type="string" />
<attribute name="unitsNOTE" value="assumes met.no FELT values are mm/6h" type="string" />
</parameter>
<parameter id="340,8,0" name="sea_ice_concentration">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="valid_min" value="0" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_ice_area_fraction" type="string" />
<attribute name="standard_name" value="sea_ice_area_fraction" type="string" />
<attribute name="units" value="1e-2" type="string" />
</parameter>
<parameter id="341,8,0" name="sea_ice_thickness">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="valid_min" value="0" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_ice_thickness" type="string" />
<attribute name="standard_name" value="sea_ice_thickness" type="string" />
<attribute name="units" value="m" type="string" />
</parameter>
<parameter id="342,6,0" name="x_ice_velocity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_ice_x_velocity" type="string" />
<attribute name="standard_name" value="sea_ice_x_velocity" type="string" />
<attribute name="units" value="m s-1" type="string" />
</parameter>
<parameter id="342,8,0" name="x_ice_velocity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_ice_x_velocity" type="string" />
<attribute name="standard_name" value="sea_ice_x_velocity" type="string" />
<attribute name="units" value="m s-1" type="string" />
</parameter>
<parameter id="343,6,0" name="y_ice_velocity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_ice_y_velocity" type="string" />
<attribute name="standard_name" value="sea_ice_y_velocity" type="string" />
<attribute name="units" value="m s-1" type="string" />
</parameter>
<parameter id="343,8,0" name="y_ice_velocity">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="sea_ice_y_velocity" type="string" />
<attribute name="standard_name" value="sea_ice_y_velocity" type="string" />
<attribute name="units" value="m s-1" type="string" />
</parameter>
<parameter id="351,8,0" name="bottom_topography">

```

```
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="valid_min" value="0" type="short" />
<attribute name="long_name" value="sea_floor_depth_below_geoid" type="string" />
<attribute name="standard_name" value="sea_floor_depth_below_geoid" type="string" />
<attribute name="units" value="m" type="string" />
</parameter>
<parameter id="371,5,0" name="nitrate">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="valid_min" value="0" type="short" />
<attribute name="scale_factor" value="6.969e-08" type="float" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="moles_of_nitrate_per_unit_mass_in_sea_water" type="string" />
<attribute name="standard_name" value="moles_of_nitrate_per_unit_mass_in_sea_water" type="string" />
<attribute name="units" value="mol kg-1" type="string" />
</parameter>
<parameter id="372,5,0" name="phosphate">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="valid_min" value="0" type="short" />
<attribute name="scale_factor" value="3.147e-08" type="float" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="moles_of_phosphate_per_unit_mass_in_sea_water" type="string" />
<attribute name="standard_name" value="moles_of_phosphate_per_unit_mass_in_sea_water" type="string" />
<attribute name="units" value="mol kg-1" type="string" />
</parameter>
<parameter id="373,5,0" name="silicate">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="valid_min" value="0" type="short" />
<attribute name="scale_factor" value="3.484e-08" type="float" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="moles_of_silicate_per_unit_mass_in_sea_water" type="string" />
<attribute name="standard_name" value="moles_of_silicate_per_unit_mass_in_sea_water" type="string" />
<attribute name="units" value="mol kg-1" type="string" />
</parameter>
<parameter id="374,5,0" name="detritus">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="valid_min" value="0" type="short" />
<attribute name="scale_factor" value="7.143e-05" type="float" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="concentration_of_detritus_in_sea_water" type="string" />
<attribute name="standard_name" value="mole_concentration_of_organic_detritus_in_sea_water_expressed_as_ni" type="string" />
<attribute name="units" value="mol m-3" type="string" />
<attribute name="units_assumption" value="units based on a density of 1025 kg m-3, not actual density" type="string" />
</parameter>
<parameter id="375,5,0" name="diatoms">
<attribute name="valid_min" value="0" type="short" />
<attribute name="scale_factor" value="7.143e-05" type="float" />
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="concentration_of_diatoms_in_sea_water" type="string" />
<attribute name="standard_name" value="mole_concentration_of_diatoms_in_sea_water_expressed_as_nitrogen" type="string" />
<attribute name="units" value="mol m-3" type="string" />
<attribute name="units_assumption" value="units based on a density of 1025 kg m-3, not actual density" type="string" />
</parameter>
<parameter id="376,5,0" name="flagellates">
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="valid_min" value="0" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="concentration_of_flagellates_in_sea_water" type="string" />
<attribute name="units" value="mgN m-3" type="string" />
</parameter>
<parameter id="377,5,0" name="oxygen">
<attribute name="valid_min" value="0" type="short" />
<attribute name="scale_factor" value="1.e-06" type="float" />
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="mass_concentration_of_oxygen_in_sea_water" type="string" />
<attribute name="standard_name" value="mass_concentration_of_oxygen_in_sea_water" type="string" />
```

```
<attribute name="units" value="kg m-3" type="string" />
</parameter>
<parameter id="378,5,0" name="silica">
<attribute name="valid_min" value="0" type="short" />
<attribute name="scale_factor" value="3.484e-08" type="float" />
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="moles_of_silica_per_unit_mass_in_sea_water" type="string" />
<attribute name="units" value="mol kg-1" type="string" />
</parameter>
<parameter id="379,5,0" name="sediment_type_1">
<attribute name="valid_min" value="0" type="short" />
<attribute name="scale_factor" value="1.e-06" type="float" />
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="mass_concentration_of_fast_sinking_material_in_sea_water" type="string" />
<attribute name="units" value="mgN m-3" type="string" />
</parameter>
<parameter id="380,5,0" name="sediment_type_2">
<attribute name="valid_min" value="0" type="short" />
<attribute name="scale_factor" value="1.e-06" type="float" />
<attribute name="_FillValue" value="-32767" type="short" />
<attribute name="cell_methods" value="time: point" type="string" />
<attribute name="long_name" value="mass_concentration_of_slow_sinking_material_in_sea_water" type="string" />
<attribute name="units" value="mgN m-3" type="string" />
</parameter>
</variables>
</cdm_felt_config>
```

Chapter 4

ncml Configuration

4.1 ncml Configuration

```

<?xml version="1.0" encoding="UTF-8"?>
<netcdf xmlns="http://www.unidata.ucar.edu/namespaces/netcdf/ncml-2.2"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.unidata.ucar.edu/namespaces/netcdf/ncml-2.2 ncml-2.2.xsd ">

<!-- rename a dimension, make sure to rename the variable-dimension, too -->
<dimension name="x_c" orgName="x" />

<!-- add/reassign a new variable attribute -->
<variable name="air_temperature">
  <attribute name="standard_name" value="temperature" type="string" />
</variable>

<!-- change the type of a variable -->
<variable name="precipitation_amount" type="short">
  <attribute name="_FillValue" value="-32767" type="short" />
  <attribute name="scale_factor" value="0.001" type="float" />
  <attribute name="add_offset" value="0" type="double" />
</variable>

<!-- rename a variable -->
<variable orgName="sea_level_pressure" name="sea_pressure" />

<!-- change units from m to km-->
<!-- make sure to put the type to change the data, too!!! -->
<variable name="x_c" type="double" orgName="x">
  <attribute name="units" value="km" type="string" />
</variable>

<variable name="surface_snow_sickness">
  <!--remove a variable-attribute -->
  <remove name="long_name" type="attribute"/>
</variable>

<!-- add/reassign a new global attribute -->
<attribute name="max_time" value="2008-05-28" type="string" />
<!-- rename a attribute -->
<!-- <attribute orgName="min_time" name="minimum_time" /> -->

<!-- remove a global attribute -->
<remove name="comment" type="attribute" />
<!-- remove a variable -->
<remove name="land_ice_area_fraction" type="variable" />

<!-- not supported -->
<group name="x" orgName="y"/>
</netcdf>

```

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

gribWriter Configuration

5.1 gribWriter Configuration

```

<?xml version="1.0" encoding="UTF-8"?>
<cdm_gribwriter_config>
<global_attributes>
<!-- type can be double, long, string -->
<attribute name="identificationOfOriginatingGeneratingCentre" value="98" type="long" /> <!-- currently used -->
<!-- use for grib1 only attributes -->
<!-- <gattribute name="" value="" type="" /> -->
<!-- use for grib2 only attributes -->
<g2attribute name="shapeOfTheEarth" value="8" type="long" /> <!-- spherical r=6,371,229.0m -->
<g2attribute name="typeOfPacking" value="grid_simple" type="string" /> <!-- jpeg2000 -->
</global_attributes>
<axes>
<!-- no vertical axis -->
<vertical_axis standard_name="">
  <!-- using height in meter, 0 -->
  <grib1 id="105" value="0" type="short"/>
  <grib2 id="103" value="0" type="short"/>
</vertical_axis>
<vertical_axis unitCompatibleTo="m">
  <!-- no standard_name for height/depth, defined only by positive=up/down -->
  <grib1 id="105" units="m" type="short"/>
  <grib2 id="103" units="m" type="double"/>
</vertical_axis>
<vertical_axis unitCompatibleTo="Pa">
  <!-- no standard_name for pressure coord, defined only by unit of hPa -->
  <grib1 id="100" units="hPa" type="short"/>
  <grib2 id="100" units="Pa" type="double"/>
</vertical_axis>
<vertical_axis standard_name="atmosphere_sigma_coordinate">
<grib1 id="107" scale_factor="1e-4" add_offset="0" type="short" />
<grib2 id="104" type="double" />
</vertical_axis>
</axes>
<variables>
<!-- translation section for parameter names/standard_names -->
<parameter standard_name="altitude">
  <grib1 parameterNumber="8" codeTable="128" units="m"/>
  <grib2 discipline="0" parameterCategory="3" parameterNumber="6" units="m"/>
</parameter>
<parameter standard_name="sea_surface_temperature">
  <grib1 parameterNumber="80" codeTable="128" units="K"/>
  <grib2 discipline="10" parameterCategory="3" parameterNumber="0" units="K"/>
</parameter>
<parameter standard_name="precipitation_amount">
  <grib1 parameterNumber="61" codeTable="128" units="kg/m2"/>
  <grib2 discipline="0" parameterCategory="1" parameterNumber="8" units="kg/m2"/>
</parameter>
<parameter standard_name="cloud_area_fraction">
<!-- total -->
  <grib1 parameterNumber="164" codeTable="128" units="1/100"/>
</parameter>
<parameter standard_name="cloud_area_fraction_in_atmosphere_layer" level="10000">
<!-- fog -->
  <grib1 parameterNumber="248" codeTable="128" units="1/100"/>
</parameter>
<parameter standard_name="cloud_area_fraction_in_atmosphere_layer" level="8500">
<!-- low clouds -->
  <grib1 parameterNumber="186" codeTable="128" units="1/100"/>
</parameter>
<parameter standard_name="cloud_area_fraction_in_atmosphere_layer" level="5000">
<!-- medium clouds -->
  <grib1 parameterNumber="187" codeTable="128" units="1/100"/>
</parameter>
<parameter standard_name="cloud_area_fraction_in_atmosphere_layer" level="3000">
<!-- high clouds -->

```

```
<grib1 parameterNumber="188" codeTable="128" units="1/100"/>
</parameter>
</variables>
</cdm_gribwriter_config>
```


Chapter 6

netcdfWriter Configuration

6.1 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.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cdm_ncwriter_config SYSTEM "cdmWriterConfig.dtd">
<cdm_ncwriter_config>

<!-- filetype are netcdf3 netcdf3_64bit netcdf4 netcdf4classic -->
<!-- compression levels are 0 (no compression) to 9 -->
<!-- <default filetype="netcdf4" compressionLevel="3" /> -->
<default filetype="netcdf3" compressionLevel="0" />

<!-- change units from m to km-->
<!-- make sure to put the type to change the data, too!!! -->
<variable name="x_c" type="double">
  <attribute name="units" value="km" type="string" />
</variable>

<!-- change the compressionLevel of a netcdf4 variable -->
<variable name="precipitation_amout" compressionLevel="3"/>

<!-- configure the output with the help of ncml -->
<ncmlConfig filename="../share/etc/ncmlCDMConfig.ncml" />

</cdm_ncwriter_config>
```


Chapter 7

Namespace Index

7.1 Namespace List

Here is a list of all namespaces with brief descriptions:

KDTree	41
MetNoFelt	45
MetNoFimex	47

Chapter 8

Class Index

8.1 Class Hierarchy

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

KDTree::_Alloc_base< _Tp, _Alloc >	55
KDTree::_Alloc_base< _Tp, _Alloc >::NoLeakAlloc	58
KDTree::_Alloc_base< _Val, _Alloc >	55
KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >	156
KDTree::_Base_iterator	59
KDTree::Iterator< _Val, _Ref, _Ptr >	62
KDTree::_Bracket_accessor< _Val >	61
KDTree::_Node_base	67
KDTree::_Node< _Val >	66
KDTree::_Node_compare< _Val, _Acc, _Cmp >	70
KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >	71
KDTree::always_true< _Tp >	75
std::binary_function< _Arg1, _Arg2, _Result > [external]	
MetNoFelt::ShortPairLess	200
MetNoFimex::CDMNameCompare	104
MetNoFimex::CachedInterpolation	76
MetNoFimex::CachedVectorReprojection	78
MetNoFimex::CDM	79
MetNoFimex::CDMNamedEntity	105
MetNoFimex::CDMAttribute	91
MetNoFimex::CDMDimension	95
MetNoFimex::CDMVariable	112
MetNoFimex::CDMReader	107
MetNoFimex::CDMExtractor	98
MetNoFimex::CDMInterpolator	101
MetNoFimex::CDMTimeInterpolator	110
MetNoFimex::FeltCDMReader	141
MetNoFimex::NcmlCDMReader	189
MetNoFimex::NetCDF_CF10_CDMReader	193
MetNoFimex::CDMWriter	115
MetNoFimex::GribApiCDMWriter	147
MetNoFimex::NetCDF_CDMWriter	191

MetNoFimex::Null_CDMWriter	195
MetNoFimex::Data	116
MetNoFimex::DataImpl< C >	121
MetNoFimex::DataTypeChanger	129
std::exception[external]	
MetNoFelt::Felt_File_Error	140
MetNoFimex::CDMException	97
MetNoFimex::UnitException	210
MetNoFelt::Felt_Array	131
MetNoFelt::Felt_File	136
MetNoFelt::FeltParameters	142
MetNoFimex::FimexTime	144
MetNoFimex::GribApiCDMWriter_ImplAbstract	152
MetNoFimex::GribApiCDMWriter_Impl1	148
MetNoFimex::GribApiCDMWriter_Impl2	150
MetNoFimex::Logger	187
MetNoFimex::ReplaceStringObject	196
MetNoFimex::ReplaceStringTimeObject	198
MetNoFimex::SpatialAxisSpec	201
KDTree::squared_difference< _Tp, _Dist >	203
KDTree::squared_difference_counted< _Tp, _Dist >	204
MetNoFimex::TimeLevelDataSliceFetcher	205
MetNoFimex::TimeSpec	206
MetNoFimex::TimeUnit	208
std::unary_function< _Arg, _Result >[external]	
MetNoFimex::CDMNameEqual	106
MetNoFimex::Units	211
MetNoFimex::XMLDoc	213

Chapter 9

Class Index

9.1 Class List

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

KDTree::_Alloc_base< _Tp, _Alloc >	55
KDTree::_Alloc_base< _Tp, _Alloc >::NoLeakAlloc	58
KDTree::_Base_iterator	59
KDTree::_Bracket_accessor< _Val >	61
KDTree::_Iterator< _Val, _Ref, _Ptr >	62
KDTree::_Node< _Val >	66
KDTree::_Node_base	67
KDTree::_Node_compare< _Val, _Acc, _Cmp >	70
KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >	71
KDTree::always_true< _Tp >	75
MetNoFimex::CachedInterpolation	76
MetNoFimex::CachedVectorReprojection	78
MetNoFimex::CDM (Data structure of the Common Data Model)	79
MetNoFimex::CDMAttribute	91
MetNoFimex::CDMDimension	95
MetNoFimex::CDMException	97
MetNoFimex::CDMExtractor	98
MetNoFimex::CDMInterpolator	101
MetNoFimex::CDMNameCompare	104
MetNoFimex::CDMNamedEntity	105
MetNoFimex::CDMNameEqual	106
MetNoFimex::CDMReader (Basic interface for CDM reading and manipulation classes)	107
MetNoFimex::CDMTimeInterpolator	110
MetNoFimex::CDMVariable	112
MetNoFimex::CDMWriter	115
MetNoFimex::Data	116
MetNoFimex::DataImpl< C >	121
MetNoFimex::DataTypeChanger	129
MetNoFelt::Felt_Array (Encapsulate parameters of a felt file)	131
MetNoFelt::Felt_File (Felt File access)	136
MetNoFelt::Felt_File_Error	140
MetNoFimex::FeltCDMReader	141
MetNoFelt::FeltParameters	142

MetNoFimex::FimexTime	144
MetNoFimex::GribApiCDMWriter	147
MetNoFimex::GribApiCDMWriter_Impl1	148
MetNoFimex::GribApiCDMWriter_Impl2	150
MetNoFimex::GribApiCDMWriter_ImplAbstract	152
KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >	156
MetNoFimex::Logger	187
MetNoFimex::Ncm1CDMReader	189
MetNoFimex::NetCDF_CDMWriter	191
MetNoFimex::NetCDF_CF10_CDMReader	193
MetNoFimex::Null_CDMWriter	195
MetNoFimex::ReplaceStringObject	196
MetNoFimex::ReplaceStringTimeObject	198
MetNoFimex::ShortPairLess	200
MetNoFimex::SpatialAxisSpec	201
KDTree::squared_difference< _Tp, _Dist >	203
KDTree::squared_difference_counted< _Tp, _Dist >	204
MetNoFimex::TimeLevelDataSliceFetcher (Read a slice of a given time/level combination from a cdmReader)	205
MetNoFimex::TimeSpec	206
MetNoFimex::TimeUnit	208
MetNoFimex::UnitException	210
MetNoFimex::Units	211
MetNoFimex::XMLDoc	213

Chapter 10

File Index

10.1 File List

Here is a list of all files with brief descriptions:

include/fimex/CacheInterpolation.h	216
include/fimex/CacheVectorReprojection.h	217
include/fimex/CDM.h	218
include/fimex/CDMAttribute.h	219
include/fimex/CDMconstants.h	220
include/fimex/CDMDataType.h	221
include/fimex/CDMDimension.h	222
include/fimex/CDMException.h	223
include/fimex/CDMExtractor.h	224
include/fimex/CDMInterpolator.h	225
include/fimex/CDMNamedEntity.h	226
include/fimex/CDMReader.h	227
include/fimex/CDMTimeInterpolator.h	228
include/fimex/CDMVariable.h	229
include/fimex/CDMWriter.h	230
include/fimex/config.h	231
include/fimex/Data.h	234
include/fimex/DataImpl.h	235
include/fimex/DataTypeChanger.h	236
include/fimex/Felt_Array.h	237
include/fimex/Felt_File.h	238
include/fimex/Felt_File_Error.h	239
include/fimex/FeltCDMReader.h	240
include/fimex/FeltParameters.h	241
include/fimex/GribApiCDMWriter.h	242
include/fimex/GribApiCDMWriter_Impl1.h	243
include/fimex/GribApiCDMWriter_Impl2.h	244
include/fimex/GribApiCDMWriter_ImplAbstract.h	245
include/fimex/interpolation.h	246
include/fimex/Logger.h	256
include/fimex/NcmlCDMReader.h	257
include/fimex/NetCDF_CDMWriter.h	258
include/fimex/NetCDF_CF10_CDMReader.h	259

include/fimex/NetCDF_Utils.h	260
include/fimex/Null_CDMWriter.h	261
include/fimex/ReplaceStringObject.h	262
include/fimex/ReplaceStringTimeObject.h	263
include/fimex/SpatialAxisSpec.h	264
include/fimex/TimeLevelDataSliceFetcher.h	265
include/fimex/TimeSpec.h	266
include/fimex/TimeUnit.h	267
include/fimex/Units.h	268
include/fimex/Utils.h	269
include/fimex/XMLDoc.h	270
include/kdtree++/allocator.hpp	271
include/kdtree++/function.hpp	272
include/kdtree++/iterator.hpp	273
include/kdtree++/kdtree.hpp	274
include/kdtree++/node.hpp	276
include/kdtree++/region.hpp	277

Chapter 11

Namespace Documentation

11.1 KDTree Namespace Reference

Classes

- class [_Alloc_base](#)
- struct [_Bracket_accessor](#)
- struct [always_true](#)
- struct [squared_difference](#)
- struct [squared_difference_counted](#)
- class [_Base_iterator](#)
- class [_Iterator](#)
- class [KDTree](#)
- struct [_Node_base](#)
- struct [_Node](#)
- class [_Node_compare](#)
- struct [_Region](#)

Functions

- `template<typename _Val, typename _Ref, typename _Ptr>`
`bool operator==(_Iterator< _Val, _Ref, _Ptr > const &, _Iterator< _Val, _Ref, _Ptr > const &)`
- `template<typename _Val>`
`bool operator==(_Iterator< _Val, const _Val &, const _Val * > const &, _Iterator< _Val, _Val &, _Val * > const &)`
- `template<typename _Val>`
`bool operator==(_Iterator< _Val, _Val &, _Val * > const &, _Iterator< _Val, const _Val &, const _Val * > const &)`
- `template<typename _Val, typename _Ref, typename _Ptr>`
`bool operator!=(_Iterator< _Val, _Ref, _Ptr > const &, _Iterator< _Val, _Ref, _Ptr > const &)`
- `template<typename _Val>`
`bool operator!=(_Iterator< _Val, const _Val &, const _Val * > const &, _Iterator< _Val, _Val &, _Val * > const &)`
- `template<typename _Val>`
`bool operator!=(_Iterator< _Val, _Val &, _Val * > const &, _Iterator< _Val, const _Val &, const _Val * > const &)`

- `template<typename _ValA, typename _ValB, typename _Cmp, typename _Acc>`
`bool _S_node_compare (const size_t __dim, const _Cmp &__cmp, const _Acc &__acc, const _ValA &__a, const _ValB &__b)`
- `template<typename _ValA, typename _ValB, typename _Dist, typename _Acc>`
`_Dist::distance_type _S_node_distance (const size_t __dim, const _Dist &__dist, const _Acc &__acc, const _ValA &__a, const _ValB &__b)`
- `template<typename _ValA, typename _ValB, typename _Dist, typename _Acc>`
`_Dist::distance_type _S_accumulate_node_distance (const size_t __dim, const _Dist &__dist, const _Acc &__acc, const _ValA &__a, const _ValB &__b)`
- `template<typename _Val, typename _Cmp, typename _Acc>`
`_Node_base * _S_node_descend (const size_t __dim, const _Cmp &__cmp, const _Acc &__acc, const _Val &__val, const _Node_base *__node)`
- `template<class SearchVal, typename _Val, typename _Cmp, typename _Acc, typename _Dist, typename _Predicate>`
`std::pair< const _Node< _Val > *, std::pair< size_t, typename _Dist::distance_type > > _S_node_nearest (const size_t __k, size_t __dim, SearchVal const &__val, const _Node< _Val > *__node, const _Node_base *__end, const _Node< _Val > *__best, typename _Dist::distance_type __max, const _Cmp &__cmp, const _Acc &__acc, const _Dist &__dist, _Predicate __p)`

11.1.1 Function Documentation

11.1.1.1 `template<typename _ValA, typename _ValB, typename _Dist, typename _Acc>`
`_Dist::distance_type KDTree::S_accumulate_node_distance (const size_t __dim, const _Dist & __dist, const _Acc & __acc, const _ValA & __a, const _ValB & __b) [inline]`

Compute the distance between two values and accumulate the result for all dimensions.

The distance functor and the accessor are references to the template parameters of the [KDTree](#).

Referenced by `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_nearest()`, and `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_nearest_if()`.

11.1.1.2 `template<typename _ValA, typename _ValB, typename _Cmp, typename _Acc> bool`
`KDTree::S_node_compare (const size_t __dim, const _Cmp & __cmp, const _Acc & __acc, const _ValA & __a, const _ValB & __b) [inline]`

Compare two values on the same dimension using a comparison functor `_Cmp` and an accessor `_Acc`.

The comparison functor and the accessor are references to the template parameters of the [KDTree](#).

Referenced by `_S_node_descend()`, and `_S_node_nearest()`.

11.1.1.3 `template<typename _Val, typename _Cmp, typename _Acc> _Node_base*`
`KDTree::S_node_descend (const size_t __dim, const _Cmp & __cmp, const _Acc & __acc, const _Val & __val, const _Node_base *__node) [inline]`

Descend on the left or the right of the node according to the comparison between the node's value and the value.

Note:

it's the caller responsibility to check if node is NULL.

References `KDTree::_Node_base::_M_left`, `KDTree::_Node_base::_M_right`, and `_S_node_compare()`.

Referenced by `_S_node_nearest()`.

11.1.1.4 `template<typename _ValA, typename _ValB, typename _Dist, typename _Acc>
_Dist::distance_type KDTree::_S_node_distance (const size_t __dim, const _Dist &
__dist, const _Acc & __acc, const _ValA & __a, const _ValB & __b) [inline]`

Compute the distance between two values for one dimension only.

The distance functor and the accessor are references to the template parameters of the [KDTree](#).

Referenced by `_S_node_nearest()`.

11.1.1.5 `template<class SearchVal, typename _Val, typename _Cmp, typename _Acc, typename
_Dist, typename _Predicate> std::pair<const _Node<_Val>*, std::pair<size_t,
typename _Dist::distance_type> > KDTree::_S_node_nearest (const size_t __k, size_t
__dim, SearchVal const & __val, const _Node<_Val> * __node, const _Node_base *
__end, const _Node<_Val> * __best, typename _Dist::distance_type __max, const
_Cmp & __cmp, const _Acc & __acc, const _Dist & __dist, _Predicate __p) [inline]`

Find the nearest node to `__val` from `__node`

If many nodes are equidistant to `__val`, the node with the lowest memory address is returned.

Returns:

the nearest node of `__end` node if no nearest node was found for the given arguments.

References `KDTree::_Node_base::_M_left`, `KDTree::_Node_base::_M_parent`, `KDTree::_Node_base::_M_right`, `_S_node_compare()`, `_S_node_descend()`, and `_S_node_distance()`.

Referenced by `KDTree::KDTree<__K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_nearest()`, and `KDTree::KDTree<__K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_nearest_if()`.

11.1.1.6 `template<typename _Val> bool KDTree::operator!= (_Iterator<_Val, _Val &, _Val * >
const & __X, _Iterator<_Val, const _Val &, const _Val * > const & __Y) [inline]`

References `KDTree::_Base_iterator::_M_node`.

11.1.1.7 `template<typename _Val> bool KDTree::operator!= (_Iterator<_Val, const _Val &,
const _Val * > const & __X, _Iterator<_Val, _Val &, _Val * > const & __Y) [inline]`

References `KDTree::_Base_iterator::_M_node`.

11.1.1.8 `template<typename _Val, typename _Ref, typename _Ptr> bool KDTree::operator!=
(_Iterator<_Val, _Ref, _Ptr > const & __X, _Iterator<_Val, _Ref, _Ptr > const & __Y)
[inline]`

References `KDTree::_Base_iterator::_M_node`.

11.1.1.9 `template<typename _Val> bool KDTree::operator== (_Iterator<_Val, _Val &, _Val * >
const & __X, _Iterator<_Val, const _Val &, const _Val * > const & __Y) [inline]`

References `KDTree::_Base_iterator::_M_node`.

11.1.1.10 `template<typename _Val> bool KDTree::operator==(_Iterator< _Val, const _Val &, const _Val * > const & __X, _Iterator< _Val, _Val &, _Val * > const & __Y)`
[inline]

References KDTree::_Base_iterator::_M_node.

11.1.1.11 `template<typename _Val, typename _Ref, typename _Ptr> bool KDTree::operator==(_Iterator< _Val, _Ref, _Ptr > const & __X, _Iterator< _Val, _Ref, _Ptr > const & __Y)`
[inline]

References KDTree::_Base_iterator::_M_node.

11.2 MetNoFelt Namespace Reference

Classes

- struct [ShortPairLess](#)
- class [Felt_Array](#)
encapsulate parameters of a felt file
- class [Felt_File](#)
Felt File access.
- class [Felt_File_Error](#)
- class [FeltParameters](#)

Typedefs

- typedef `set< pair< short, short >, ShortPairLess >` [ShortPairSet](#)
- typedef `map< pair< short, short >, short, ShortPairLess >` [ShortPairMap](#)

Functions

- `time_t` [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](#) ()

11.2.1 Typedef Documentation

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

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

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

set<pair<short,short> > with comparator

11.2.2 Function Documentation

11.2.2.1 `const boost::array<short, 16>& MetNoFelt::ANY_ARRAY ()`

11.2.2.2 `const boost::array<short, 20>& MetNoFelt::ANY_ARRAY20 ()`

11.2.2.3 `const int MetNoFelt::ANY_VALUE () [inline]`

11.2.2.4 `std::string MetNoFelt::getProjString (int gridType, const boost::array< float, 6 > & gridParameters) throw (Felt_File_Error)`

11.2.2.5 `pair<short, short> MetNoFelt::index16toLevelPair (const boost::array< short, 16 > & idx)`

convert the 16-short header to a levelPair

11.2.2.6 `time_t MetNoFelt::index16toTime (const boost::array< short, 16 > & idx)`

convert the 16-short header to a time

11.2.2.7 `const std::string& MetNoFelt::UNDEFINED ()`

11.3 MetNoFimex Namespace Reference

Classes

- class [CachedInterpolation](#)
- class [CachedVectorReprojection](#)
- class [CDM](#)

Data structure of the Common Data Model.

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

Basic interface for CDM reading and manipulation classes.

- class [CDMTimeInterpolator](#)
- class [CDMVariable](#)
- class [CDMWriter](#)
- class [Data](#)
- class [DataImpl](#)
- class [DataTypeChanger](#)
- class [FeltCDMReader](#)
- class [GribApiCDMWriter](#)
- class [GribApiCDMWriter_Impl1](#)
- class [GribApiCDMWriter_Impl2](#)
- class [GribApiCDMWriter_ImplAbstract](#)
- class [Logger](#)
- class [NcmlCDMReader](#)
- class [NetCDF_CDMWriter](#)
- class [NetCDF_CF10_CDMReader](#)
- class [Null_CDMWriter](#)
- class [ReplaceStringObject](#)
- class [ReplaceStringTimeObject](#)
- 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](#)
- class [XMLDoc](#)

Typedefs

- typedef boost::shared_ptr< [Logger](#) > [LoggerPtr](#)
- 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

- [std::vector](#)< [CDMAttribute](#) > [projStringToAttributes](#) ([std::string](#) projStr)
convert a proj4 string to a list of CDMAttributes usable for CF-1.0 projection variable
- [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)
- boost::shared_ptr< [Data](#) > [createData](#) ([CDMDataType](#) datatype, size_t length) 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 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, size_t length, 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 T1, typename T2>`
`boost::shared_array< T1 >` `convertArrayType` (const `boost::shared_array< T2 >` &inData, `size_t` length, `double` oldFill, `double` oldScale, `double` oldOffset, `double` newFill, `double` newScale, `double` newOffset)
- `Logger::LogLevel` `defaultLogLevel` ()
- `void` `defaultLogLevel` (`Logger::LogLevel`)
- `LoggerPtr` `getLogger` (const `std::string` &className)
- `NcType` `cdmDataType2ncType` (`CDMDataType` dt)
- `CDMDataType` `ncType2cdmDataType` (`NcType` dt)
- `boost::shared_ptr< Data >` `ncValues2Data` (`NcValues *values`, `NcType` dt, `size_t` length)
- `std::ostream &` `operator<<` (`std::ostream` &out, const `FimexTime` &fTime)
- `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=",")
- `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<typename T>`
T `string2type` (`std::string` s)
- `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)

11.3.1 Typedef Documentation

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

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

11.3.2 Enumeration Type Documentation

11.3.2.1 `enum MetNoFimex::CDMDataType`

Enumerator:

CDM_NAT

CDM_CHAR

CDM_SHORT

CDM_INT

CDM_FLOAT

CDM_DOUBLE

CDM_STRING

11.3.3 Function Documentation

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

convert attributes of a projection-variable to a projString

Parameters:

attrs attributes of the projection variable

Returns:

proj4 string

11.3.3.2 `NcType MetNoFimex::cdmDataType2ncType (CDMDataType dt)`

conversion from CDMDataType to NcType

11.3.3.3 `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

11.3.3.4 `template<typename T1, typename T2> boost::shared_array<T1> MetNoFimex::convertArrayType (const boost::shared_array< T2 > & inData, size_t length, double oldFill, double oldScale, double oldOffset, double newFill, double newScale, double newOffset) [inline]`

References fill().

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

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

Parameters:

datatype

size_t length of the data array

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, and CDM_SHORT.

11.3.3.6 `boost::shared_ptr<Data> MetNoFimex::createData (CDMDataType datatype, size_t length) throw (CDMException)`

create a Data-pointer of the datatype

Parameters:

datatype

size_t length of the data array

Returns:

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

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

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

Parameters:

datatype

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

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

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

11.3.3.10 `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.

11.3.3.11 `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

11.3.3.12 `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

11.3.3.13 `std::string MetNoFimex::getXmlName (const xmlNodePtr node)`

a memory-save form of `xmlGetName`

Returns:

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

11.3.3.14 `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

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

11.3.3.16 `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()`.

11.3.3.17 `CDMDataType MetNoFimex::ncType2cdmDataType (NcType dt)`

conversion from `NcType` to `CDMDataType`

11.3.3.18 `boost::shared_ptr<Data> MetNoFimex::ncValues2Data (NcValues * values, NcType dt, size_t length)`

convert `ncValues` to a `Data` pointer

Warning:

: the data belonging to `values` will be freed within this function or with the `shared_array`. Do not free the values otherwise!

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

11.3.3.20 `std::vector<CDMAAttribute> MetNoFimex::projStringToAttributes (std::string projStr)`

convert a proj4 string to a list of CDMAAttributes usable for CF-1.0 projection variable

currently, projStrings of the form +proj=[stere] +lat_0=? +lon_0=? +lat_ts=?

11.3.3.21 `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

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

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

Round a double to integer.

11.3.3.23 `CDMDataType MetNoFimex::string2datatype (const std::string & s)`

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

11.3.3.24 `FimexTime MetNoFimex::string2FimexTime (const std::string & str) throw (CDMException)`

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

convert a string to lowercase

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

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

Tokenize a string by a delimiter. This function will automaticall 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().

11.3.3.28 `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()`.

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

Remove leading and trailing spaces.

Parameters:

str string to trim

Referenced by tokenizeDotted().

11.3.3.30 `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()`.

Chapter 12

Class Documentation

12.1 KDTree::_Alloc_base< _Tp, _Alloc > Class Template Reference

```
#include <allocator.hpp>
```

Public Types

- typedef [_Node< _Tp > _Node_](#)
- typedef [_Node::_Base_ptr _Base_ptr](#)
- typedef [_Alloc allocator_type](#)

Public Member Functions

- [_Alloc_base \(allocator_type const &__A\)](#)
- [allocator_type get_allocator \(\) const](#)

Protected Member Functions

- [_Node_ * _M_allocate_node \(\)](#)
- [void _M_deallocate_node \(_Node_ *const __P\)](#)
- [void _M_construct_node \(_Node_ *__p, _Tp const __V=_Tp\(\), _Base_ptr const __PARENT=NULL, _Base_ptr const __LEFT=NULL, _Base_ptr const __RIGHT=NULL\)](#)
- [void _M_destroy_node \(_Node_ *__p\)](#)

Protected Attributes

- [allocator_type _M_node_allocator](#)

Classes

- class [NoLeakAlloc](#)

`template<typename _Tp, typename _Alloc> class KDTree::_Alloc_base< _Tp, _Alloc >`

12.1.1 Member Typedef Documentation

12.1.1.1 `template<typename _Tp, typename _Alloc> typedef _Node<_Tp>
KDTree::_Alloc_base< _Tp, _Alloc >::_Node_`

12.1.1.2 `template<typename _Tp, typename _Alloc> typedef _Node::_Base_ptr
KDTree::_Alloc_base< _Tp, _Alloc >::_Base_ptr`

Reimplemented in [KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >](#).

12.1.1.3 `template<typename _Tp, typename _Alloc> typedef _Alloc KDTree::_Alloc_base< _Tp,
_Alloc >::allocator_type`

Reimplemented in [KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >](#).

12.1.2 Constructor & Destructor Documentation

12.1.2.1 `template<typename _Tp, typename _Alloc> KDTree::_Alloc_base< _Tp, _Alloc
>::_Alloc_base (allocator_type const & __A) [inline]`

12.1.3 Member Function Documentation

12.1.3.1 `template<typename _Tp, typename _Alloc> allocator_type KDTree::_Alloc_base< _Tp,
_Alloc >::get_allocator () const [inline]`

Reimplemented in [KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >](#).

12.1.3.2 `template<typename _Tp, typename _Alloc> _Node* KDTree::_Alloc_base< _Tp,
_Alloc >::_M_allocate_node () [inline, protected]`

12.1.3.3 `template<typename _Tp, typename _Alloc> void KDTree::_Alloc_base< _Tp, _Alloc
>::_M_deallocate_node (_Node_*const __P) [inline, protected]`

Referenced by [KDTree::_Alloc_base< _Tp, _Alloc >::NoLeakAlloc::~~NoLeakAlloc\(\)](#).

12.1.3.4 `template<typename _Tp, typename _Alloc> void KDTree::_Alloc_base<_Tp, _Alloc>::_M_construct_node(_Node_* __p, _Tp const __V = _Tp(), _Base_ptr const __PARENT = NULL, _Base_ptr const __LEFT = NULL, _Base_ptr const __RIGHT = NULL) [inline, protected]`

12.1.3.5 `template<typename _Tp, typename _Alloc> void KDTree::_Alloc_base<_Tp, _Alloc>::_M_destroy_node(_Node_* __p) [inline, protected]`

12.1.4 Member Data Documentation

12.1.4.1 `template<typename _Tp, typename _Alloc> allocator_type KDTree::_Alloc_base<_Tp, _Alloc>::_M_node_allocator [protected]`

Referenced by `KDTree::_Alloc_base<_Val, _Alloc>::_M_allocate_node()`, `KDTree::_Alloc_base<_Val, _Alloc>::_M_deallocate_node()`, `KDTree::_Alloc_base<_Val, _Alloc>::_M_destroy_node()`, and `KDTree::_Alloc_base<_Val, _Alloc>::get_allocator()`.

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

- `include/kdtree++/allocator.hpp`

12.2 KDTree::_Alloc_base< _Tp, _Alloc >::NoLeakAlloc Class Reference

```
#include <allocator.hpp>
```

Public Member Functions

- [NoLeakAlloc](#) ([_Alloc_base](#) *b)
- [_Node_](#) * [get](#) ()
- void [disconnect](#) ()
- [~NoLeakAlloc](#) ()

```
template<typename _Tp, typename _Alloc> class KDTree::_Alloc_base< _Tp, _Alloc >::NoLeakAlloc
```

12.2.1 Constructor & Destructor Documentation

12.2.1.1 `template<typename _Tp, typename _Alloc> KDTree::_Alloc_base< _Tp, _Alloc >::NoLeakAlloc::NoLeakAlloc (_Alloc_base * b)` [[inline](#)]

12.2.1.2 `template<typename _Tp, typename _Alloc> KDTree::_Alloc_base< _Tp, _Alloc >::NoLeakAlloc::~NoLeakAlloc ()` [[inline](#)]

References [KDTree::_Alloc_base< _Tp, _Alloc >::_M_deallocate_node\(\)](#).

12.2.2 Member Function Documentation

12.2.2.1 `template<typename _Tp, typename _Alloc> _Node_* KDTree::_Alloc_base< _Tp, _Alloc >::NoLeakAlloc::get ()` [[inline](#)]

12.2.2.2 `template<typename _Tp, typename _Alloc> void KDTree::_Alloc_base< _Tp, _Alloc >::NoLeakAlloc::disconnect ()` [[inline](#)]

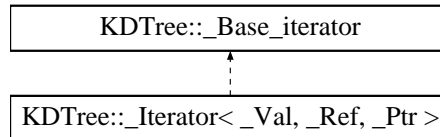
The documentation for this class was generated from the following file:

- [include/kdtree++/allocator.hpp](#)

12.3 KDTree::_Base_iterator Class Reference

```
#include <iterator.hpp>
```

Inheritance diagram for KDTree::_Base_iterator::



Protected Types

- typedef [_Node_base::_Base_const_ptr](#) [_Base_const_ptr](#)

Protected Member Functions

- [_Base_iterator](#) ([_Base_const_ptr](#) const __N=NULL)
- [_Base_iterator](#) ([_Base_iterator](#) const &__THAT)
- void [_M_increment](#) ()
- void [_M_decrement](#) ()

Protected Attributes

- [_Base_const_ptr](#) [_M_node](#)

Friends

- class [KDTree](#)

12.3.1 Member Typedef Documentation

12.3.1.1 typedef [_Node_base::_Base_const_ptr](#) [KDTree::_Base_iterator::_Base_const_ptr](#) [protected]

12.3.2 Constructor & Destructor Documentation

12.3.2.1 [KDTree::_Base_iterator::_Base_iterator](#) ([_Base_const_ptr](#) const __N = NULL) [inline, protected]

12.3.2.2 [KDTree::_Base_iterator::_Base_iterator](#) ([_Base_iterator](#) const & __THAT) [inline, protected]

12.3.3 Member Function Documentation

12.3.3.1 void [KDTree::_Base_iterator::_M_increment](#) () [inline, protected]

References [_M_node](#).

Referenced by `KDTree::_Iterator<_Val, _Ref, _Ptr>::operator++()`.

12.3.3.2 `void KDTree::_Base_iterator::_M_decrement()` [`inline`, `protected`]

References `_M_node`.

Referenced by `KDTree::_Iterator<_Val, _Ref, _Ptr>::operator--()`.

12.3.4 Friends And Related Function Documentation

12.3.4.1 `friend class KDTree` [`friend`]

12.3.5 Member Data Documentation

12.3.5.1 `_Base_const_ptr KDTree::_Base_iterator::_M_node` [`protected`]

Referenced by `_M_decrement()`, `_M_increment()`, `KDTree::_Iterator<_Val, _Ref, _Ptr>::get_raw_node()`, `KDTree::operator!=()`, `KDTree::_Iterator<_Val, _Ref, _Ptr>::operator*`, and `KDTree::operator==()`.

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

- [include/kdtree++/iterator.hpp](#)

12.4 KDTree::_Bracket_accessor<_Val> Struct Template Reference

```
#include <function.hpp>
```

Public Types

- typedef _Val::value_type [result_type](#)

Public Member Functions

- [result_type operator\(\)](#) (_Val const &V, size_t const N) const

```
template<typename _Val> struct KDTree::_Bracket_accessor<_Val>
```

12.4.1 Member Typedef Documentation

12.4.1.1 `template<typename _Val> typedef _Val::value_type KDTree::_Bracket_accessor<_Val>::result_type`

12.4.2 Member Function Documentation

12.4.2.1 `template<typename _Val> result_type KDTree::_Bracket_accessor<_Val>::operator()(_Val const & V, size_t const N) const` [*inline*]

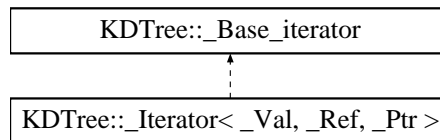
The documentation for this struct was generated from the following file:

- `include/kdtree++/function.hpp`

12.5 KDTree::_Iterator< _Val, _Ref, _Ptr > Class Template Reference

```
#include <iterator.hpp>
```

Inheritance diagram for KDTree::_Iterator< _Val, _Ref, _Ptr >::



Public Types

- typedef `_Val` [value_type](#)
- typedef `_Ref` [reference](#)
- typedef `_Ptr` [pointer](#)
- typedef `_Iterator< _Val, _Val &, _Val * >` [iterator](#)
- typedef `_Iterator< _Val, _Val const &, _Val const * >` [const_iterator](#)
- typedef `_Iterator< _Val, _Ref, _Ptr >` [_Self](#)
- typedef `_Node< _Val > const *` [_Link_const_type](#)
- typedef `std::bidirectional_iterator_tag` [iterator_category](#)
- typedef `ptrdiff_t` [difference_type](#)

Public Member Functions

- `_Iterator` ()
- `_Iterator` ([_Link_const_type](#) const __N)
- `_Iterator` ([iterator](#) const &__THAT)
- `_Link_const_type` `get_raw_node` () const
- `reference` `operator*` () const
- `pointer` `operator →` () const
- `_Self` `operator++` ()
- `_Self` `operator++` (int)
- `_Self` & `operator--` ()
- `_Self` `operator--` (int)

Friends

- bool `operator==` (`_Iterator< _Val, _Ref, _Ptr >` const &, `_Iterator< _Val, _Ref, _Ptr >` const &)
- bool `operator==` (`_Iterator< _Val, const _Val &, const _Val * >` const &, `_Iterator< _Val, _Val &, _Val * >` const &)
- bool `operator==` (`_Iterator< _Val, _Val &, _Val * >` const &, `_Iterator< _Val, const _Val &, const _Val * >` const &)
- bool `operator!=` (`_Iterator< _Val, _Ref, _Ptr >` const &, `_Iterator< _Val, _Ref, _Ptr >` const &)
- bool `operator!=` (`_Iterator< _Val, const _Val &, const _Val * >` const &, `_Iterator< _Val, _Val &, _Val * >` const &)
- bool `operator!=` (`_Iterator< _Val, _Val &, _Val * >` const &, `_Iterator< _Val, const _Val &, const _Val * >` const &)

```
template<typename _Val, typename _Ref, typename _Ptr> class KDTree::_Iterator<_Val, _Ref,
_Ptr >
```

12.5.1 Member Typedef Documentation

12.5.1.1 `template<typename _Val, typename _Ref, typename _Ptr> typedef _Val
KDTree::_Iterator<_Val, _Ref, _Ptr >::value_type`

12.5.1.2 `template<typename _Val, typename _Ref, typename _Ptr> typedef _Ref
KDTree::_Iterator<_Val, _Ref, _Ptr >::reference`

12.5.1.3 `template<typename _Val, typename _Ref, typename _Ptr> typedef _Ptr
KDTree::_Iterator<_Val, _Ref, _Ptr >::pointer`

12.5.1.4 `template<typename _Val, typename _Ref, typename _Ptr> typedef _Iterator<_Val,
_Val&, _Val*> KDTree::_Iterator<_Val, _Ref, _Ptr >::iterator`

12.5.1.5 `template<typename _Val, typename _Ref, typename _Ptr> typedef _Iterator<_Val, _Val
const&, _Val const*> KDTree::_Iterator<_Val, _Ref, _Ptr >::const_iterator`

12.5.1.6 `template<typename _Val, typename _Ref, typename _Ptr> typedef _Iterator<_Val, _Ref,
_Ptr> KDTree::_Iterator<_Val, _Ref, _Ptr >::_Self`

12.5.1.7 `template<typename _Val, typename _Ref, typename _Ptr> typedef _Node<_Val> const*
KDTree::_Iterator<_Val, _Ref, _Ptr >::_Link_const_type`

12.5.1.8 `template<typename _Val, typename _Ref, typename _Ptr> typedef
std::bidirectional_iterator_tag KDTree::_Iterator<_Val, _Ref, _Ptr >::iterator_category`

12.5.1.9 `template<typename _Val, typename _Ref, typename _Ptr> typedef ptrdiff_t
KDTree::_Iterator<_Val, _Ref, _Ptr >::difference_type`

12.5.2 Constructor & Destructor Documentation

12.5.2.1 `template<typename _Val, typename _Ref, typename _Ptr> KDTree::_Iterator<_Val,
_Ref, _Ptr >::_Iterator () [inline]`

12.5.2.2 `template<typename _Val, typename _Ref, typename _Ptr> KDTree::_Iterator<_Val,
_Ref, _Ptr >::_Iterator (_Link_const_type const __N) [inline]`

12.5.2.3 `template<typename _Val, typename _Ref, typename _Ptr> KDTree::_Iterator<_Val,
_Ref, _Ptr >::_Iterator (iterator const & __THAT) [inline]`

12.5.3 Member Function Documentation

12.5.3.1 `template<typename _Val, typename _Ref, typename _Ptr> _Link_const_type
KDTree::_Iterator<_Val, _Ref, _Ptr >::get_raw_node () const [inline]`

References KDTree::_Base_iterator::_M_node.

Referenced by KDTree::KDTree<__K, _Val, _Acc, _Dist, _Cmp, _Alloc >::erase().

12.5.3.2 `template<typename _Val, typename _Ref, typename _Ptr> reference
KDTree::_Iterator< _Val, _Ref, _Ptr >::operator* () const [inline]`

References `KDTree::_Base_iterator::_M_node`.

Referenced by `KDTree::_Iterator< _Val, _Ref, _Ptr >::operator → ()`.

12.5.3.3 `template<typename _Val, typename _Ref, typename _Ptr> pointer KDTree::_Iterator<
_Val, _Ref, _Ptr >::operator → () const [inline]`

References `KDTree::_Iterator< _Val, _Ref, _Ptr >::operator*()`.

12.5.3.4 `template<typename _Val, typename _Ref, typename _Ptr> _Self KDTree::_Iterator<
_Val, _Ref, _Ptr >::operator++ () [inline]`

References `KDTree::_Base_iterator::_M_increment()`.

12.5.3.5 `template<typename _Val, typename _Ref, typename _Ptr> _Self KDTree::_Iterator<
_Val, _Ref, _Ptr >::operator++ (int) [inline]`

References `KDTree::_Base_iterator::_M_increment()`.

12.5.3.6 `template<typename _Val, typename _Ref, typename _Ptr> _Self& KDTree::_Iterator<
_Val, _Ref, _Ptr >::operator- () [inline]`

References `KDTree::_Base_iterator::_M_decrement()`.

12.5.3.7 `template<typename _Val, typename _Ref, typename _Ptr> _Self KDTree::_Iterator<
_Val, _Ref, _Ptr >::operator- (int) [inline]`

References `KDTree::_Base_iterator::_M_decrement()`.

12.5.4 Friends And Related Function Documentation

12.5.4.1 `template<typename _Val, typename _Ref, typename _Ptr> bool operator==(_Iterator<_Val, _Ref, _Ptr> const & __X, _Iterator<_Val, _Ref, _Ptr> const & __Y) [friend]`

12.5.4.2 `template<typename _Val, typename _Ref, typename _Ptr> bool operator==(_Iterator<_Val, const _Val &, const _Val * > const & __X, _Iterator<_Val, _Val &, _Val * > const & __Y) [friend]`

12.5.4.3 `template<typename _Val, typename _Ref, typename _Ptr> bool operator==(_Iterator<_Val, _Val &, _Val * > const & __X, _Iterator<_Val, const _Val &, const _Val * > const & __Y) [friend]`

12.5.4.4 `template<typename _Val, typename _Ref, typename _Ptr> bool operator!=(_Iterator<_Val, _Ref, _Ptr> const & __X, _Iterator<_Val, _Ref, _Ptr> const & __Y) [friend]`

12.5.4.5 `template<typename _Val, typename _Ref, typename _Ptr> bool operator!=(_Iterator<_Val, const _Val &, const _Val * > const & __X, _Iterator<_Val, _Val &, _Val * > const & __Y) [friend]`

12.5.4.6 `template<typename _Val, typename _Ref, typename _Ptr> bool operator!=(_Iterator<_Val, _Val &, _Val * > const & __X, _Iterator<_Val, const _Val &, const _Val * > const & __Y) [friend]`

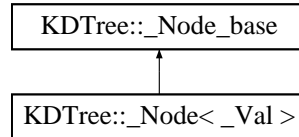
The documentation for this class was generated from the following file:

- [include/kdtree++/iterator.hpp](#)

12.6 KDTree::_Node< _Val > Struct Template Reference

```
#include <node.hpp>
```

Inheritance diagram for KDTree::_Node< _Val >::



Public Types

- typedef [_Node](#) * [_Link_type](#)

Public Member Functions

- [_Node](#) ([_Val](#) const &__VALUE=[_Val](#)(), [_Base_ptr](#) const __PARENT=NULL, [_Base_ptr](#) const __LEFT=NULL, [_Base_ptr](#) const __RIGHT=NULL)

Public Attributes

- [_Val](#) [_M_value](#)

```
template<typename _Val> struct KDTree::_Node< _Val >
```

12.6.1 Member Typedef Documentation

12.6.1.1 `template<typename _Val> typedef _Node* KDTree::_Node< _Val >::_Link_type`

12.6.2 Constructor & Destructor Documentation

12.6.2.1 `template<typename _Val> KDTree::_Node< _Val >::_Node (_Val const & __VALUE = _Val (), _Base_ptr const __PARENT = NULL, _Base_ptr const __LEFT = NULL, _Base_ptr const __RIGHT = NULL) [inline]`

12.6.3 Member Data Documentation

12.6.3.1 `template<typename _Val> _Val KDTree::_Node< _Val >::_M_value`

Referenced by `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_check_children()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_find()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_find_exact()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_insert()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_matches_node_in_d()`, and `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_value()`.

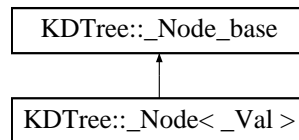
The documentation for this struct was generated from the following file:

- `include/kdtree++/node.hpp`

12.7 KDTree::_Node_base Struct Reference

```
#include <node.hpp>
```

Inheritance diagram for KDTree::_Node_base::



Public Types

- typedef [_Node_base](#) * [_Base_ptr](#)
- typedef [_Node_base](#) const * [_Base_const_ptr](#)

Public Member Functions

- [_Node_base](#) ([_Base_ptr](#) const __PARENT=NULL, [_Base_ptr](#) const __LEFT=NULL, [_Base_ptr](#) const __RIGHT=NULL)

Static Public Member Functions

- static [_Base_ptr](#) [_S_minimum](#) ([_Base_ptr](#) __x)
- static [_Base_ptr](#) [_S_maximum](#) ([_Base_ptr](#) __x)

Public Attributes

- [_Base_ptr](#) [_M_parent](#)
- [_Base_ptr](#) [_M_left](#)
- [_Base_ptr](#) [_M_right](#)

12.7.1 Member Typedef Documentation

12.7.1.1 `typedef _Node_base* KDTree::_Node_base::_Base_ptr`

12.7.1.2 `typedef _Node_base const* KDTree::_Node_base::_Base_const_ptr`

12.7.2 Constructor & Destructor Documentation

12.7.2.1 `KDTree::_Node_base::_Node_base (_Base_ptr const __PARENT = NULL, _Base_ptr const __LEFT = NULL, _Base_ptr const __RIGHT = NULL) [inline]`

12.7.3 Member Function Documentation

12.7.3.1 `static _Base_ptr KDTree::_Node_base::_S_minimum (_Base_ptr __x) [inline, static]`

References `_M_left`.

Referenced by `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_minimum()`.

12.7.3.2 `static _Base_ptr KDTree::_Node_base::_S_maximum (_Base_ptr __x) [inline, static]`

References `_M_right`.

Referenced by `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_maximum()`.

12.7.4 Member Data Documentation

12.7.4.1 `_Base_ptr KDTree::_Node_base::_M_parent`

Referenced by `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_empty_initialise()`, `KDTree::_S_node_nearest()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_parent()`, and `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_set_parent()`.

12.7.4.2 `_Base_ptr KDTree::_Node_base::_M_left`

Referenced by `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_leftmost()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_set_leftmost()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_left()`, `_S_minimum()`, `KDTree::_S_node_descend()`, `KDTree::_S_node_nearest()`, and `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_set_left()`.

12.7.4.3 `_Base_ptr KDTree::_Node_base::_M_right`

Referenced by `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_rightmost()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_set_rightmost()`, `_S_maximum()`, `KDTree::_S_node_descend()`, `KDTree::_S_node_nearest()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_right()`, and `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_set_right()`.

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

- [include/kdtree++/node.hpp](#)

12.8 KDTree::_Node_compare< _Val, _Acc, _Cmp > Class Template Reference

```
#include <node.hpp>
```

Public Member Functions

- [_Node_compare](#) (size_t const __DIM, _Acc const &acc, _Cmp const &cmp)
- bool [operator\(\)](#) (_Val const &__A, _Val const &__B) const

```
template<typename _Val, typename _Acc, typename _Cmp> class KDTree::_Node_compare< _Val, _Acc, _Cmp >
```

12.8.1 Constructor & Destructor Documentation

12.8.1.1 `template<typename _Val, typename _Acc, typename _Cmp> KDTree::_Node_compare< _Val, _Acc, _Cmp >::_Node_compare (size_t const __DIM, _Acc const & acc, _Cmp const & cmp) [inline]`

12.8.2 Member Function Documentation

12.8.2.1 `template<typename _Val, typename _Acc, typename _Cmp> bool KDTree::_Node_compare< _Val, _Acc, _Cmp >::operator() (_Val const & __A, _Val const & __B) const [inline]`

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

- `include/kdtree++/node.hpp`

12.9 KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp > Struct Template Reference

```
#include <region.hpp>
```

Public Types

- typedef `_Val` [value_type](#)
- typedef `_SubVal` [subvalue_type](#)
- typedef `std::pair< _Region, _SubVal >` [_CenterPt](#)

Public Member Functions

- [_Region](#) (`_Acc` const &__acc=`_Acc`(), const `_Cmp` &__cmp=`_Cmp`())
- `template<typename Val>`
[_Region](#) (`Val` const &__V, `_Acc` const &__acc=`_Acc`(), const `_Cmp` &__cmp=`_Cmp`())
- `template<typename Val>`
[_Region](#) (`Val` const &__V, [subvalue_type](#) const &__R, `_Acc` const &__acc=`_Acc`(), const `_Cmp` &__cmp=`_Cmp`())
- `bool` [intersects_with](#) (`_CenterPt` const &__THAT) const
- `bool` [intersects_with](#) (`_Region` const &__THAT) const
- `bool` [encloses](#) ([value_type](#) const &__V) const
- `_Region` & [set_high_bound](#) ([value_type](#) const &__V, `size_t` const __L)
- `_Region` & [set_low_bound](#) ([value_type](#) const &__V, `size_t` const __L)

Public Attributes

- `subvalue_type` [_M_low_bounds](#) [`__K`]
- `subvalue_type` [_M_high_bounds](#) [`__K`]
- `_Acc` [_M_acc](#)
- `_Cmp` [_M_cmp](#)

```
template<size_t const __K, typename _Val, typename _SubVal, typename _Acc, typename _Cmp>
struct KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >
```

12.9.1 Member Typedef Documentation

12.9.1.1 `template<size_t const __K, typename _Val, typename _SubVal, typename _Acc, typename _Cmp> typedef _Val KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::value_type`

12.9.1.2 `template<size_t const __K, typename _Val, typename _SubVal, typename _Acc, typename _Cmp> typedef _SubVal KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::subvalue_type`

12.9.1.3 `template<size_t const __K, typename _Val, typename _SubVal, typename _Acc, typename _Cmp> typedef std::pair<_Region, SubVal> KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_CenterPt`

12.9.2 Constructor & Destructor Documentation

12.9.2.1 `template<size_t const __K, typename _Val, typename _SubVal, typename _Acc, typename _Cmp> KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_Region (_Acc const & __acc = _Acc (), const _Cmp & __cmp = _Cmp ()) [inline]`

12.9.2.2 `template<size_t const __K, typename _Val, typename _SubVal, typename _Acc, typename _Cmp> template<typename Val> KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_Region (Val const & __V, _Acc const & __acc = _Acc (), const _Cmp & __cmp = _Cmp ()) [inline]`

References `KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_M_acc`, `KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_M_high_bounds`, and `KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_M_low_bounds`.

12.9.2.3 `template<size_t const __K, typename _Val, typename _SubVal, typename _Acc, typename _Cmp> template<typename Val> KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_Region (Val const & __V, subvalue_type const & __R, _Acc const & __acc = _Acc (), const _Cmp & __cmp = _Cmp ()) [inline]`

References `KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_M_acc`, `KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_M_high_bounds`, and `KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_M_low_bounds`.

12.9.3 Member Function Documentation

12.9.3.1 `template<size_t const __K, typename _Val, typename _SubVal, typename _Acc, typename _Cmp> bool KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::intersects_with (_CenterPt const & __THAT) const [inline]`

References `KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_M_cmp`, `KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_M_high_bounds`, `KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_M_low_bounds`, `std::pair<_T1, _T2 >::first`, and `std::pair<_T1, _T2 >::second`.

Referenced by `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_count_within_range()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_find_within_range()`, and

KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_visit_within_range().

12.9.3.2 `template<size_t const __K, typename _Val, typename _SubVal, typename _Acc, typename _Cmp> bool KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::intersects_with (_Region< __K, _Val, _SubVal, _Acc, _Cmp > const & __THAT) const` [inline]

References KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_M_cmp, KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_M_high_bounds, and KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_M_low_bounds.

12.9.3.3 `template<size_t const __K, typename _Val, typename _SubVal, typename _Acc, typename _Cmp> bool KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::encloses (value_type const & __V) const` [inline]

References KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_M_acc, KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_M_cmp, KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_M_high_bounds, and KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_M_low_bounds.

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_count_within_range(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_find_within_range(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_visit_within_range().

12.9.3.4 `template<size_t const __K, typename _Val, typename _SubVal, typename _Acc, typename _Cmp> _Region& KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::set_high_bound (value_type const & __V, size_t const __L)` [inline]

References KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_M_acc, and KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_M_high_bounds.

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_count_within_range(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_find_within_range(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_visit_within_range().

12.9.3.5 `template<size_t const __K, typename _Val, typename _SubVal, typename _Acc, typename _Cmp> _Region& KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::set_low_bound (value_type const & __V, size_t const __L)` [inline]

References KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_M_acc, and KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_M_low_bounds.

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_count_within_range(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_find_within_range(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_visit_within_range().

12.9.4 Member Data Documentation

12.9.4.1 `template<size_t const __K, typename _Val, typename _SubVal, typename _Acc, typename _Cmp> subvalue_type KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_M_low_bounds[__K]`

Referenced by KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_Region(), KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::encloses(), KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp

>::intersects_with(), and KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::set_low_bound().

12.9.4.2 `template<size_t const __K, typename _Val, typename _SubVal, typename _Acc, typename _Cmp> subvalue_type KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_M_high_bounds[__K]`

Referenced by KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_Region(), KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::encloses(), KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::intersects_with(), and KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::set_high_bound().

12.9.4.3 `template<size_t const __K, typename _Val, typename _SubVal, typename _Acc, typename _Cmp> _Acc KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_M_acc`

Referenced by KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_Region(), KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::encloses(), KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::set_high_bound(), and KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::set_low_bound().

12.9.4.4 `template<size_t const __K, typename _Val, typename _SubVal, typename _Acc, typename _Cmp> _Cmp KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::_M_cmp`

Referenced by KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::encloses(), and KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::intersects_with().

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

- [include/kdtree++/region.hpp](#)

12.10 KDTree::always_true< _Tp > Struct Template Reference

```
#include <function.hpp>
```

Public Member Functions

- `bool operator() (const _Tp &) const`

```
template<typename _Tp> struct KDTree::always_true< _Tp >
```

12.10.1 Member Function Documentation

12.10.1.1 `template<typename _Tp> bool KDTree::always_true< _Tp >::operator() (const _Tp &) const` [`inline`]

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

- `include/kdtree++/function.hpp`

12.11 MetNoFimex::CachedInterpolation Class Reference

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

Public Member Functions

- [CachedInterpolation](#) ()
- [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](#) ()
- boost::shared_array< float > [interpolateValues](#) (boost::shared_array< float > *inData*, size_t *size*, size_t &*newSize*)

12.11.1 Detailed Description

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

12.11.2 Constructor & Destructor Documentation

12.11.2.1 [MetNoFimex::CachedInterpolation::CachedInterpolation](#) () [inline]

12.11.2.2 [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

12.11.2.3 [virtual MetNoFimex::CachedInterpolation::~~CachedInterpolation](#) () [inline, virtual]

12.11.3 Member Function Documentation

12.11.3.1 [boost::shared_array<float> MetNoFimex::CachedInterpolation::interpolateValues](#) (boost::shared_array< float > *inData*, size_t *size*, size_t &*newSize*)

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

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

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

12.12 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

12.12.1 Constructor & Destructor Documentation

12.12.1.1 [MetNoFimex::CachedVectorReprojection::CachedVectorReprojection](#) () [inline]

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

12.12.1.3 virtual [MetNoFimex::CachedVectorReprojection::~~CachedVectorReprojection](#) () [inline, virtual]

12.12.2 Member Function Documentation

12.12.2.1 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

12.12.2.2 size_t [MetNoFimex::CachedVectorReprojection::getXSize](#) () const [inline]

12.12.2.3 size_t [MetNoFimex::CachedVectorReprojection::getYSize](#) () const [inline]

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

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

12.13 MetNoFimex::CDM Class Reference

Data structure of the Common Data Model.

```
#include <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 ()`
- virtual `~CDM ()`
- 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
- 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")
- 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
- **AttrVec** [getProjection](#) (**std::string** varName) const

get the projection attributes (as of CF-1.0) 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

12.13.1 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>

12.13.2 Member Typedef Documentation

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

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

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

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

12.13.3 Constructor & Destructor Documentation

12.13.3.1 `MetNoFimex::CDM::CDM ()`

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

12.13.4 Member Function Documentation

12.13.4.1 `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

12.13.4.2 `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

12.13.4.3 `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

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

test if variable exists

Parameters:

varName name of variable

12.13.4.5 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:

copies of the attributes matching the request

12.13.4.6 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:

copies of the attributes matching the request

12.13.4.7 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

12.13.4.8 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 regex the string-value of the attribute will match against

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

remove a variable and corresponding attributes

Parameters:

variableName the variable to remove

12.13.4.10 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

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

check if the dimension exists

Parameters:

dimName name of the dimension

12.13.4.12 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

12.13.4.13 const CDMDimension& MetNoFimex::CDM::getDimension (const std::string & *dimName*) const throw (CDMException)**12.13.4.14 bool MetNoFimex::CDM::testDimensionInUse (const std::string & *name*) const**

test if a dimension is actively in use

Parameters:

name dimensionName

12.13.4.15 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

12.13.4.16 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 use in a variable

12.13.4.17 `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](#)

12.13.4.18 `bool MetNoFimex::CDM::hasUnlimitedDim (const CDMVariable & var) const`

test if a variable contains the unlimited dim

Returns:

true/false

12.13.4.19 `void MetNoFimex::CDM::addAttribute (const std::string & varName, const CDMAAttribute & attr) throw (CDMException)`

add an attribute to `cdm`

Parameters:

varName name of the variable the attribute belongs to

attr the [CDMAAttribute](#)

Exceptions:

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

12.13.4.20 `void MetNoFimex::CDM::addOrReplaceAttribute (const std::string & varName, const CDMAAttribute & attr) throw (CDMException)`

add or replace an attribute of the `cdm`

Parameters:

varName name of variable the attribute belongs to

attr the [CDMAAttribute](#)

Exceptions:

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

12.13.4.21 `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 [CDMAAttribute](#)

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

print a xml representation to the stream

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

the namespace for global attributes

12.13.4.24 const DimVec& MetNoFimex::CDM::getDimensions () const [inline]

get the dimension

12.13.4.25 const VarVec& MetNoFimex::CDM::getVariables () const [inline]

get the variables

12.13.4.26 const StrAttrVecMap& MetNoFimex::CDM::getAttributes () const [inline]

get the attributes

Returns:

map of type <variableName <attributeName, attribute>>

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

get the attributes of an variable

Parameters:

varName name of variable

12.13.4.28 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

12.13.4.29 `const CDMAAttribute& 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

12.13.4.30 `bool MetNoFimex::CDM::getAttribute (const std::string & varName, const std::string & attrName, CDMAAttribute & 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

12.13.4.31 `double MetNoFimex::CDM::getFillValue (const std::string & varName) const`

get the fill value of an variable (_FillValue attribute)

Returns:

value of _FillValue attribute, or MIFL_UNDEFINED_F

12.13.4.32 `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

12.13.4.33 `bool MetNoFimex::CDM::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

12.13.4.34 `AttrVec MetNoFimex::CDM::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

12.13.4.35 `std::string MetNoFimex::CDM::getHorizontalXAxis (std::string varName) const`

get the x-(lon) axis of the variable

Parameters:

varName name of variable

Returns:

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

12.13.4.36 `std::string MetNoFimex::CDM::getHorizontalYAxis (std::string varName) const`

get the y-(lat) axis of the variable

Parameters:

varName name of variable

Returns:

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

12.13.4.37 `bool MetNoFimex::CDM::getLatitudeLongitude (std::string varName, std::string & latitude, std::string & longitude) const`

detect the latitude and longitude coordinates of the variable

Detect the the latitude and longitude coordinates of the variable, this might be one of the dimensions, or a multi-dimensional field of lat(x,y) lon(x,y) variables

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

12.13.4.38 `std::string MetNoFimex::CDM::getTimeAxis (std::string varName) const`

get the time axis of the variable

Parameters:

varName name of variable

Returns:

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

12.13.4.39 `std::string MetNoFimex::CDM::getVerticalAxis (std::string varName) const`

get the vertical axis of the variable

Parameters:

varName name of variable

Returns:

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

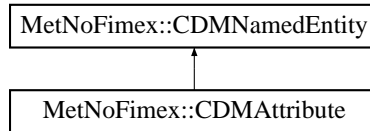
The documentation for this class was generated from the following file:

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

12.14 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

12.14.1 Constructor & Destructor Documentation

12.14.1.1 **MetNoFimex::CDMAtribute::CDMAtribute** ()

12.14.1.2 **MetNoFimex::CDMAtribute::CDMAtribute** (**std::string** name, **std::string** value) [explicit]

create a string attribute

12.14.1.3 **MetNoFimex::CDMAtribute::CDMAtribute** (**std::string** name, **char** value) [explicit]

create a char attribute with a char array of length 1

12.14.1.4 **MetNoFimex::CDMAtribute::CDMAtribute** (**std::string** name, **int** value) [explicit]

create a int attribute with a int array of length 1

12.14.1.5 **MetNoFimex::CDMAtribute::CDMAtribute** (**std::string** name, **short** value) [explicit]

create a short attribute with a short array of length 1

12.14.1.6 **MetNoFimex::CDMAtribute::CDMAtribute** (**std::string** name, **float** value) [explicit]

create a float attribute with a float array of length 1

12.14.1.7 **MetNoFimex::CDMAtribute::CDMAtribute** (**std::string** name, **double** value) [explicit]

create a double attribute with a double array of length 1

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

create a attribute with the low level information

12.14.1.9 `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

12.14.1.10 `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

12.14.1.11 `virtual MetNoFimex::CDMAAttribute::~~CDMAAttribute () [virtual]`

12.14.2 Member Function Documentation

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

retrieve the name of the attribute

Implements [MetNoFimex::CDMNamedEntity](#).

12.14.2.2 `void MetNoFimex::CDMAAttribute::setName (std::string newName) [inline]`

set the name of the attribute

12.14.2.3 `const std::string MetNoFimex::CDMAAttribute::getStringValue () const`

retrieve the stringified value of the attribute

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

retrieve the data-pointer of the attribute

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

set the data for this attribute

12.14.2.6 `const CDMDatatype MetNoFimex::CDMAAttribute::getDataType () const [inline]`

retrieve the datatype of the attribute

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

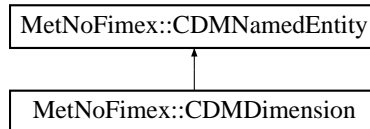
The documentation for this class was generated from the following file:

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

12.15 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

12.15.1 Constructor & Destructor Documentation

12.15.1.1 [MetNoFimex::CDMDimension::CDMDimension \(\)](#)

12.15.1.2 [MetNoFimex::CDMDimension::CDMDimension \(std::string name, long length\)](#)

12.15.1.3 [virtual MetNoFimex::CDMDimension::~~CDMDimension \(\)](#) [virtual]

12.15.2 Member Function Documentation

12.15.2.1 [const std::string& MetNoFimex::CDMDimension::getName \(\) const](#) [inline, virtual]

Implements [MetNoFimex::CDMNamedEntity](#).

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

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

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

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

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

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

print xml representation to stream

Parameters:

out stream to write xml to

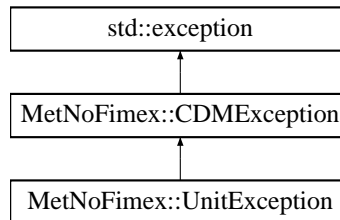
The documentation for this class was generated from the following file:

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

12.16 MetNoFimex::CDMException Class Reference

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

Inheritance diagram for MetNoFimex::CDMException::



Public Member Functions

- [CDMException \(\)](#)
- [CDMException \(const std::string &msg\)](#)
- [CDMException \(const CDMException &rhs\) throw \(\)](#)
- [CDMException & operator= \(const CDMException &rhs\) throw \(\)](#)
- [virtual ~CDMException \(\) throw \(\)](#)
- [virtual const char * what \(\) const throw \(\)](#)

12.16.1 Constructor & Destructor Documentation

12.16.1.1 [MetNoFimex::CDMException::CDMException \(\)](#) [inline]

12.16.1.2 [MetNoFimex::CDMException::CDMException \(const std::string & msg\)](#) [inline, explicit]

12.16.1.3 [MetNoFimex::CDMException::CDMException \(const CDMException & rhs\) throw \(\)](#) [inline]

12.16.1.4 [virtual MetNoFimex::CDMException::~~CDMException \(\) throw \(\)](#) [inline, virtual]

12.16.2 Member Function Documentation

12.16.2.1 [CDMException& MetNoFimex::CDMException::operator= \(const CDMException & rhs\) throw \(\)](#) [inline]

12.16.2.2 [virtual const char* MetNoFimex::CDMException::what \(\) const throw \(\)](#) [inline, virtual]

Reimplemented from `std::exception`.

References `std::basic_string<_CharT, _Traits, _Alloc >::c_str()`.

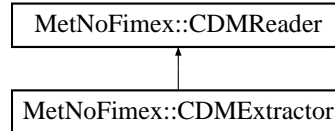
The documentation for this class was generated from the following file:

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

12.17 MetNoFimex::CDMExtractor Class Reference

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

Inheritance diagram for MetNoFimex::CDMExtractor::



Public Member Functions

- [CDMExtractor](#) (boost::shared_ptr< [CDMReader](#) > dataReader)
- virtual [~CDMExtractor](#) ()
- virtual const 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

12.17.1 Constructor & Destructor Documentation

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

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

12.17.2 Member Function Documentation

12.17.2.1 virtual const 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 function 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](#).

12.17.2.2 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

12.17.2.3 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

12.17.2.4 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

12.17.2.5 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

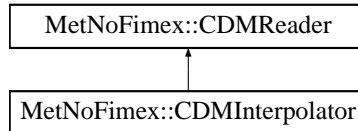
The documentation for this class was generated from the following file:

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

12.18 MetNoFimex::CDMInterpolator Class Reference

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

Inheritance diagram for MetNoFimex::CDMInterpolator::



Public Member Functions

- [CDMInterpolator](#) (boost::shared_ptr< [CDMReader](#) > dataReader)
- virtual [~CDMInterpolator](#) ()
- virtual const 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

12.18.1 Constructor & Destructor Documentation

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

12.18.1.2 virtual [MetNoFimex::CDMInterpolator::~~CDMInterpolator](#) () [virtual]

12.18.2 Member Function Documentation

12.18.2.1 virtual const 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

12.18.2.2 `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

12.18.2.3 `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

12.18.2.4 `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

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

Returns:

the name used for latitude in the automatic coordinate generation

12.18.2.6 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:

lonName name for longitude

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

the name used for longitude in the automatic coordinate generation

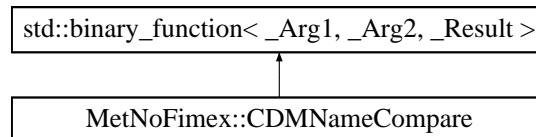
The documentation for this class was generated from the following file:

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

12.19 MetNoFimex::CDMNameCompare Struct Reference

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

Inheritance diagram for MetNoFimex::CDMNameCompare::



Public Member Functions

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

12.19.1 Detailed Description

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

12.19.2 Member Function Documentation

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

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

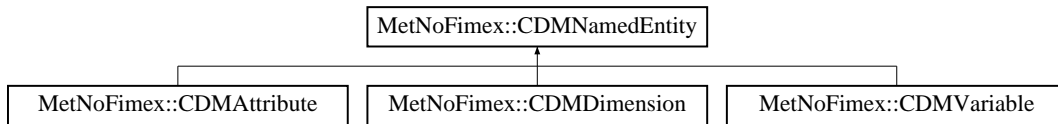
The documentation for this struct was generated from the following file:

- `include/fimex/CDMNamedEntity.h`

12.20 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

12.20.1 Detailed Description

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

12.20.2 Constructor & Destructor Documentation

12.20.2.1 virtual [MetNoFimex::CDMNamedEntity::~CDMNamedEntity](#) () [pure virtual]

12.20.3 Member Function Documentation

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

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

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

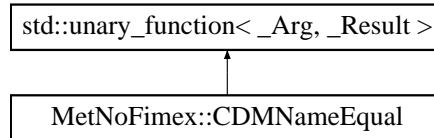
The documentation for this class was generated from the following file:

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

12.21 MetNoFimex::CDMNameEqual Class Reference

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

Inheritance diagram for MetNoFimex::CDMNameEqual::



Public Member Functions

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

12.21.1 Detailed Description

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

12.21.2 Constructor & Destructor Documentation

12.21.2.1 [MetNoFimex::CDMNameEqual::CDMNameEqual](#) (**std::string** *name*) [[inline](#), [explicit](#)]

12.21.2.2 [MetNoFimex::CDMNameEqual::CDMNameEqual](#) ([CDMNamedEntity](#) & *entity*) [[inline](#), [explicit](#)]

12.21.2.3 [MetNoFimex::CDMNameEqual::~~CDMNameEqual](#) () [[inline](#)]

12.21.3 Member Function Documentation

12.21.3.1 **bool** [MetNoFimex::CDMNameEqual::operator\(\)](#) (const [CDMNamedEntity](#) & *e*) [[inline](#)]

References [MetNoFimex::CDMNamedEntity::getName\(\)](#).

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

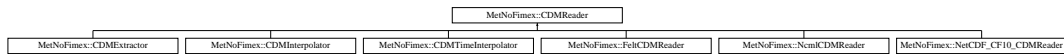
- [include/fimex/CDMNamedEntity.h](#)

12.22 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 const 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 const boost::shared_ptr< [Data](#) > [getData](#) (const **std::string** &varName) throw (CDMException)
 - data-reading function to be called from the [CDMWriter](#)*
- virtual const boost::shared_ptr< [Data](#) > [getScaledDataSlice](#) (const **std::string** &varName, size_t unLimDimPos) throw (CDMException)
 - read and scale a dataslice*
- virtual const boost::shared_ptr< [Data](#) > [getScaledData](#) (const **std::string** &varName) throw (CDMException)
 - read and scale the complete data*

Protected Member Functions

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

Protected Attributes

- [CDM](#) [cdm](#)

12.22.1 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](#)

12.22.2 Constructor & Destructor Documentation

12.22.2.1 `MetNoFimex::CDMReader::CDMReader ()` [inline]

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

12.22.3 Member Function Documentation

12.22.3.1 `virtual const CDM& MetNoFimex::CDMReader::getCDM () const` [inline, virtual]

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

References `cdm`.

12.22.3.2 `virtual const 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 function 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::CDMExtractor](#), [MetNoFimex::CDMTimeInterpolator](#), [MetNoFimex::Ncm1CDMReader](#), and [MetNoFimex::NetCDF_CF10_CDMReader](#).

12.22.3.3 `virtual const 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

12.22.3.4 `virtual const 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

12.22.3.5 `virtual const 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

12.22.3.6 `virtual const 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

12.22.4 Member Data Documentation

12.22.4.1 `CDM MetNoFimex::CDMReader::cdm [protected]`

Referenced by `MetNoFimex::FeltCDMReader::getCDM()`, and `getCDM()`.

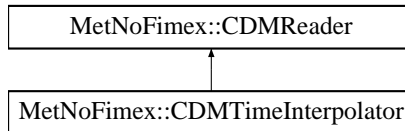
The documentation for this class was generated from the following file:

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

12.23 MetNoFimex::CDMTimeInterpolator Class Reference

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

Inheritance diagram for MetNoFimex::CDMTimeInterpolator::



Public Member Functions

- `CDMTimeInterpolator` (boost::shared_ptr< `CDMReader` > `dataReader`)
- virtual `~CDMTimeInterpolator` ()
- virtual const 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`)

12.23.1 Constructor & Destructor Documentation

12.23.1.1 `MetNoFimex::CDMTimeInterpolator::CDMTimeInterpolator` (boost::shared_ptr< `CDMReader` > `dataReader`)

12.23.1.2 virtual `MetNoFimex::CDMTimeInterpolator::~~CDMTimeInterpolator` () [virtual]

12.23.2 Member Function Documentation

12.23.2.1 virtual const 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`

Implements `MetNoFimex::CDMReader`.

12.23.2.2 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

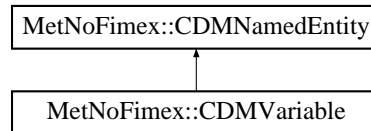
The documentation for this class was generated from the following file:

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

12.24 MetNoFimex::CDMVariable Class Reference

```
#include <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 data from this variable
- int [hasData](#) () const
check if real data has been set with [setData\(\)](#) (null-pointer reference returns false)

12.24.1 Constructor & Destructor Documentation

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

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

12.24.2 Member Function Documentation

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

Implements [MetNoFimex::CDMNamedEntity](#).

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

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

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

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

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

12.24.2.7 `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"

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

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

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

get the spatial counterpart of this vector

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

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

12.24.2.11 `bool MetNoFimex::CDMVariable::checkDimension (const std::string & dimension) const`

check the dimension of a variable

Parameters:

dimension the dimension to check for

12.24.2.12 `void MetNoFimex::CDMVariable::toXMLStream (std::ostream & out) const`

print a xml representation to the stream without attributes

12.24.2.13 `void MetNoFimex::CDMVariable::toXMLStream (std::ostream & out, const std::vector< CDMAAttribute > & attrs) const`

print a xml representation to the stream with attributes

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

add data to the variable

12.24.2.15 `const boost::shared_ptr<Data> MetNoFimex::CDMVariable::getData () const [inline]`

retrieve 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

12.24.2.16 `int MetNoFimex::CDMVariable::hasData () const [inline]`

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

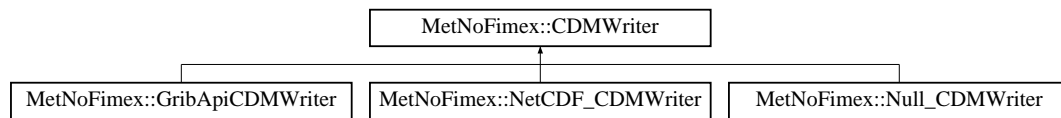
The documentation for this class was generated from the following file:

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

12.25 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*

12.25.1 Constructor & Destructor Documentation

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

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

12.25.2 Member Data Documentation

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

12.25.2.2 const **std::string** [MetNoFimex::CDMWriter::outputFile](#) [[protected](#)]

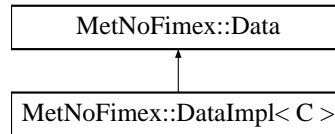
The documentation for this class was generated from the following file:

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

12.26 MetNoFimex::Data Class Reference

```
#include <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 last=-1)=0` throw (CDMEException)
- 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 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`

12.26.1 Detailed Description

General class for storing different basic array pointers plus length

12.26.2 Constructor & Destructor Documentation

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

12.26.3 Member Function Documentation

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

size of the data

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

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

sizeof the data-impl datatype

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

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

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

12.26.3.4 `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 >](#).

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

retrieve data as char

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

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

retrieve data as char

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

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

retrieve data as short

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

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

retrieve data as short

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

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

retrieve data as int

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

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

retrieve data as int

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

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

retrieve data as float

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

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

retrieve data as float (eventually copy)

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

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

retrieve data as double

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

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

retrieve data as double

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

12.26.3.15 `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 >](#).

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

set a value at the desired position

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

12.26.3.17 `virtual void MetNoFimex::Data::setValues (size_t startPos, const Data & data, size_t first = 0, size_t last = -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

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

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

12.26.3.18 `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 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 ($Product(orgDimSize) \neq size$)

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

12.26.3.19 `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 >](#).

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

return the CDMDataType of this data

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

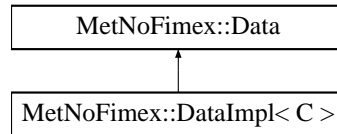
The documentation for this class was generated from the following file:

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

12.27 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
sizeof 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 boost::shared_ptr< [Data](#) > [slice](#) (**std::vector**< size_t > orgDimSize, **std::vector**< size_t > startDims, **std::vector**< size_t > outputDimSize) throw (CDMEException)
get a 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 first, InputIterator last, 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 >`

12.27.1 Constructor & Destructor Documentation

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

constructor where the array will be automatically allocated

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

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

12.27.2 Member Function Documentation

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

size of the data

Implements [MetNoFimex::Data](#).

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

sizeof of the data-impl datatype

Implements [MetNoFimex::Data](#).

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

Implements [MetNoFimex::Data](#).

12.27.2.4 `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()`.

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

get the datapointer of the data

12.27.2.6 `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

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

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

retrieve data as char

Implements [MetNoFimex::Data](#).

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

retrieve data as char

Implements [MetNoFimex::Data](#).

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

retrieve data as short

Implements [MetNoFimex::Data](#).

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

retrieve data as short

Implements [MetNoFimex::Data](#).

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

retrieve data as int

Implements [MetNoFimex::Data](#).

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

retrieve data as int

Implements [MetNoFimex::Data](#).

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

retrieve data as float

Implements [MetNoFimex::Data](#).

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

retrieve data as float (eventually copy)

Implements [MetNoFimex::Data](#).

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

retrieve data as double

Implements [MetNoFimex::Data](#).

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

retrieve data as double

Implements [MetNoFimex::Data](#).

12.27.2.18 `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()`.

12.27.2.19 `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](#).

12.27.2.20 `template<typename C> void MetNoFimex::DataImpl< C >::setValues (size_t
startPos, const Data & data, size_t first = 0, size_t last = -1) 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

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

Implements [MetNoFimex::Data](#).

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

12.27.2.22 `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](#).

12.27.2.23 `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](#).

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

set the values of the data by the input-iterator

References `MetNoFimex::type2string()`.

12.27.2.25 `template<> void MetNoFimex::DataImpl< char >::setValues (size_t startPos, const Data & data, size_t first, size_t last) 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

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

Implements [MetNoFimex::Data](#).

12.27.2.26 `template<> void MetNoFimex::DataImpl< short >::setValues (size_t startPos, const Data & data, size_t first, size_t last) 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

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

Implements [MetNoFimex::Data](#).

12.27.2.27 `template<> void MetNoFimex::DataImpl< int >::setValues (size_t startPos, const Data & data, size_t first, size_t last) 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

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

Implements [MetNoFimex::Data](#).

12.27.2.28 `template<> void MetNoFimex::DataImpl< float >::setValues (size_t startPos, const Data & data, size_t first, size_t last) 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

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

Implements [MetNoFimex::Data](#).

12.27.2.29 `template<> void MetNoFimex::DataImpl< double >::setValues (size_t startPos, const Data & data, size_t first, size_t last) 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

last 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](#)

12.28 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

12.28.1 Detailed Description

brief wrapper class around data->convertType

12.28.2 Constructor & Destructor Documentation

12.28.2.1 MetNoFimex::DataTypeChanger::DataTypeChanger (CDMDataType *oldType*) [explicit]

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

12.28.2.2 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.

12.28.2.3 `virtual MetNoFimex::DataTypeChanger::~~DataTypeChanger ()` [virtual]

12.28.3 Member Function Documentation

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

convert the data to the new scale/fill/offset

12.28.3.2 `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](#)

12.29 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**< time_t > [getTimes](#) () const
- **vector**< short > [getLevels](#) () const
- **vector**< **pair**< short, short > > [getLevelPairs](#) () const
- void [addIdent19](#) (time_t time, **pair**< short, short > levelPair, short value)
- short [getIdent19](#) (time_t time, **pair**< short, short > levelPair) const throw (Felt_File_Error)
- short [getIdent19](#) (**pair**< short, short > levelPair) const throw (Felt_File_Error)
- short [getIdent19](#) (time_t 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](#) (time_t time, short level) throw (Felt_File_Error)
- int [getFieldSize](#) (time_t time, short level) const throw (Felt_File_Error)

12.29.1 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

12.29.2 Constructor & Destructor Documentation

12.29.2.1 MetNoFelt::Felt_Array::Felt_Array ()

constructor

12.29.2.2 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 always return 1 for float and double

12.29.2.3 virtual MetNoFelt::Felt_Array::~~Felt_Array () [virtual]

12.29.3 Member Function Documentation

12.29.3.1 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

12.29.3.2 const boost::array<short, 16>& MetNoFelt::Felt_Array::getIndexHeader () const [inline]

get the time/level independent index of this header

12.29.3.3 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

12.29.3.4 const boost::array<short, 20>& MetNoFelt::Felt_Array::getDataHeader () const
[inline]

get the time/level independent data-header

12.29.3.5 short MetNoFelt::Felt_Array::getLevelType () const [inline]

get the felt level type of this array

12.29.3.6 void MetNoFelt::Felt_Array::setGridType (int gridType) [inline]

set the gridType as used in libmi gridPar function

12.29.3.7 int MetNoFelt::Felt_Array::getGridType () const [inline]

get the gridType

12.29.3.8 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

12.29.3.9 const boost::array<float, 6>& MetNoFelt::Felt_Array::getGridParameters () const
[inline]

get the extra grid information from the end of the data

12.29.3.10 const string& MetNoFelt::Felt_Array::getName () const

return the parameter name

12.29.3.11 const string& MetNoFelt::Felt_Array::getDatatype () const [inline]

return the datatype as string short|float|double

12.29.3.12 double MetNoFelt::Felt_Array::getFillValue () const [inline]

return the changed fill used in Felt_File::getScaledDataSlice

12.29.3.13 void MetNoFelt::Felt_Array::setFillValue (double *fillValue*) [inline]

set the fill value to be used in Felt_File::getScaledDataSlice

12.29.3.14 vector<time_t> MetNoFelt::Felt_Array::getTimes () const

return the times available for this parameter, sorted

12.29.3.15 vector<short> MetNoFelt::Felt_Array::getLevels () const

return the levels available for this parameter, sorted

12.29.3.16 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

12.29.3.17 void MetNoFelt::Felt_Array::addIdent19 (time_t *time*, pair< short, short > *levelPair*, short *value*) [inline]

add the ident19 parameter from the data-header

12.29.3.18 short MetNoFelt::Felt_Array::getIdent19 (time_t *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

12.29.3.19 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

12.29.3.20 short MetNoFelt::Felt_Array::getIdent19 (time_t *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

12.29.3.21 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

12.29.3.22 int MetNoFelt::Felt_Array::getX () const [inline]

return x/longitude size

12.29.3.23 int MetNoFelt::Felt_Array::getY () const [inline]

return y/latitude size

12.29.3.24 short MetNoFelt::Felt_Array::getVerticalFeltType () const [inline]

return the felt-type of the vertical axis

12.29.3.25 double MetNoFelt::Felt_Array::getScalingFactor () const

return scalingFactor

12.29.3.26 boost::array<short, 16> const MetNoFelt::Felt_Array::getIndex (time_t time, short level) throw (Felt_File_Error)

return a copy of the index used within this [Felt_Array](#)

12.29.3.27 int MetNoFelt::Felt_Array::getFieldSize (time_t time, short level) const throw (Felt_File_Error)

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

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

12.30 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 **std::time_t** time, const short level) throw (Felt_File_Error)
retrieve a data slice
- boost::shared_ptr< [MetNoFimex::Data](#) > [getScaledDataSlice](#) (const **std::string** &compName, const **std::time_t** 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**< **time_t** > [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

12.30.1 Detailed Description

Felt File access.

[Felt_File](#) gives c++ style access to felt files. It uses internally libmi and caches the table of contents

12.30.2 Constructor & Destructor Documentation

12.30.2.1 MetNoFelt::Felt_File::Felt_File () [inline]

constructor

open an empty felt file, just a default constructor, no useful information

12.30.2.2 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

12.30.2.3 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`

12.30.2.4 virtual MetNoFelt::Felt_File::~~Felt_File () [virtual]

12.30.3 Member Function Documentation

12.30.3.1 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

12.30.3.2 `std::vector<short> MetNoFelt::Felt_File::getDataSlice (const std::string & compName, const std::time_t 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

12.30.3.3 `boost::shared_ptr<MetNoFimex::Data> MetNoFelt::Felt_File::getScaledDataSlice (const std::string & compName, const std::time_t 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

12.30.3.4 `std::vector<Felt_Array> MetNoFelt::Felt_File::listFeltArrays ()`

retrieve all felt arrays

12.30.3.5 `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

12.30.3.6 `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

12.30.3.7 `const ShortPairMap& MetNoFelt::Felt_File::getHybridLevels () const` `[inline]`

12.30.3.8 `std::vector<time_t> MetNoFelt::Felt_File::getFeltTimes () const`

all time values, sorted

12.30.3.9 int MetNoFelt::Felt_File::getNX () const

get size in x direction

12.30.3.10 int MetNoFelt::Felt_File::getNY () const

get size in y direction

**12.30.3.11 boost::shared_ptr<MetNoFimex::Data> MetNoFelt::Felt_File::getXData () const
throw (Felt_File_Error)**

get the values of the x axis

**12.30.3.12 boost::shared_ptr<MetNoFimex::Data> MetNoFelt::Felt_File::getYData () const
throw (Felt_File_Error)**

get the values of the y axis

12.30.3.13 short MetNoFelt::Felt_File::getGridType () const throw (Felt_File_Error)

assumes one set of grid-type for the whole file

**12.30.3.14 const boost::array<float, 6>& MetNoFelt::Felt_File::getGridParameters () const
throw (Felt_File_Error)**

assumes one set of grid-parameters for the whole file

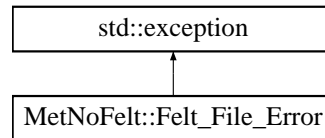
The documentation for this class was generated from the following file:

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

12.31 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)
- virtual `~Felt_File_Error` () throw ()
- virtual const char * `what` () const throw ()

12.31.1 Constructor & Destructor Documentation

12.31.1.1 `MetNoFelt::Felt_File_Error::Felt_File_Error (const std::string & message)`
[explicit]

12.31.1.2 `virtual MetNoFelt::Felt_File_Error::~~Felt_File_Error () throw ()` [virtual]

12.31.2 Member Function Documentation

12.31.2.1 `virtual const char* MetNoFelt::Felt_File_Error::what () const throw ()` [virtual]

Reimplemented from `std::exception`.

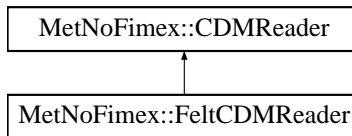
The documentation for this class was generated from the following file:

- `include/fimex/Felt_File_Error.h`

12.32 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 const boost::shared_ptr< [Data](#) > [getDataSlice](#) (const **std::string** &varName, size_t unLimDimPos) throw (CDMException)
- virtual const [CDM](#) & [getCDM](#) () const

12.32.1 Constructor & Destructor Documentation

12.32.1.1 [MetNoFimex::FeltCDMReader::FeltCDMReader](#) (**std::string** *filename*, **std::string** *configFilename*) throw (CDMException)

12.32.1.2 virtual [MetNoFimex::FeltCDMReader::~~FeltCDMReader](#) () [virtual]

12.32.2 Member Function Documentation

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

12.32.2.2 virtual const [CDM](#)& [MetNoFimex::FeltCDMReader::getCDM](#) () const [inline, virtual]

Reimplemented from [MetNoFimex::CDMReader](#).

References [MetNoFimex::CDMReader::cdm](#).

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

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

12.33 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](#) ()

12.33.1 Constructor & Destructor Documentation

12.33.1.1 MetNoFelt::FeltParameters::FeltParameters ()

12.33.1.2 MetNoFelt::FeltParameters::FeltParameters (std::string *filename*) [explicit]

initialize all known felt parameters from a diana-setup file

Parameters:

filename diana setup file

12.33.1.3 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

12.33.1.4 **virtual** MetNoFelt::FeltParameters::~~FeltParameters () [virtual]

12.33.2 Member Function Documentation

12.33.2.1 **const** boost::array<short, 16>& MetNoFelt::FeltParameters::getParameters (const std::string &)

12.33.2.2 **const** std::string& MetNoFelt::FeltParameters::getParameterName (const boost::array< short, 16 > &)

12.33.2.3 **std::string** MetNoFelt::FeltParameters::getParameterDatatype (const std::string & *parameterName*) **const**

12.33.2.4 **double** MetNoFelt::FeltParameters::getParameterFillValue (const std::string & *parameterName*) **const**

12.33.2.5 **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](#)

12.34 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)

12.34.1 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

12.34.2 Member Function Documentation

12.34.2.1 `bool MetNoFimex::FimexTime::operator==(const FimexTime & rhs) const`

compare two fimexTimes

12.34.2.2 `bool MetNoFimex::FimexTime::operator!=(const FimexTime & rhs) const` [inline]

compare two fimexTimes

12.34.2.3 `bool MetNoFimex::FimexTime::operator>(const FimexTime & rhs) const` [inline]

compare two fimexTimes

References toLong().

12.34.2.4 `bool MetNoFimex::FimexTime::operator<(const FimexTime & rhs) const` [inline]

compare two fimexTimes

12.34.2.5 `bool MetNoFimex::FimexTime::operator>=(const FimexTime & rhs) const` [inline]

compare two fimexTimes

12.34.2.6 `bool MetNoFimex::FimexTime::operator<=(const FimexTime & rhs) const` [inline]

compare two fimexTimes

12.34.3 Member Data Documentation

12.34.3.1 `unsigned short MetNoFimex::FimexTime::msecond`

millisecond

12.34.3.2 `char MetNoFimex::FimexTime::second`

second (0-59)

12.34.3.3 char MetNoFimex::FimexTime::minute

minute (0-59)

12.34.3.4 char MetNoFimex::FimexTime::hour

hour (0-23)

12.34.3.5 char MetNoFimex::FimexTime::mday

day of month (1-31)

12.34.3.6 char MetNoFimex::FimexTime::month

month (1-12)

12.34.3.7 unsigned short MetNoFimex::FimexTime::year

year (2008 as of writing)

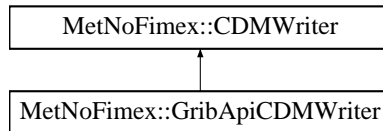
The documentation for this class was generated from the following file:

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

12.35 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](#) ()

12.35.1 Constructor & Destructor Documentation

12.35.1.1 [MetNoFimex::GribApiCDMWriter::GribApiCDMWriter](#) (const boost::shared_ptr< [CDMReader](#) > *cdmReader*, const **std::string** & *outputFile*, const int *gribVersion*, const **std::string** & *configFile*)

12.35.1.2 [virtual MetNoFimex::GribApiCDMWriter::~~GribApiCDMWriter](#) () [virtual]

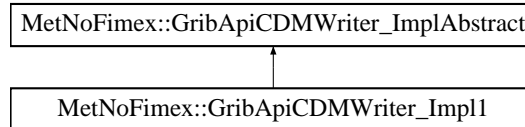
The documentation for this class was generated from the following file:

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

12.36 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)

12.36.1 Detailed Description

Implementaionn of a writer using GribApi for grib1

12.36.2 Constructor & Destructor Documentation

12.36.2.1 [MetNoFimex::GribApiCDMWriter_Impl1::GribApiCDMWriter_Impl1](#) (const boost::shared_ptr< [CDMReader](#) > &cdmReader, const std::string &outputFile, const std::string &configFile)

12.36.2.2 [virtual MetNoFimex::GribApiCDMWriter_Impl1::~~GribApiCDMWriter_Impl1](#) ()
[virtual]

12.36.3 Member Function Documentation

12.36.3.1 [virtual void MetNoFimex::GribApiCDMWriter_Impl1::setParameter](#) (const std::string &varName, const [FimexTime](#) &fTime, double levelValue) throw (CDMException)
[virtual]

Implements [MetNoFimex::GribApiCDMWriter_ImplAbstract](#).

12.36.3.2 [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](#).

12.36.3.3 virtual void MetNoFimex::GribApiCDMWriter_Impl1::setLevel (const std::string & varName, double levelValue) [virtual]

Implements [MetNoFimex::GribApiCDMWriter_ImplAbstract](#).

12.36.3.4 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](#).

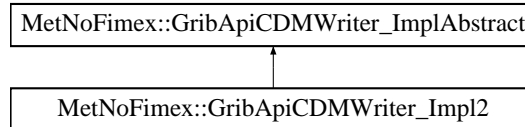
The documentation for this class was generated from the following file:

- [include/fimex/GribApiCDMWriter_Impl1.h](#)

12.37 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)

12.37.1 Detailed Description

Implementaionn of a writer using GribApi for grib2

12.37.2 Constructor & Destructor Documentation

12.37.2.1 [MetNoFimex::GribApiCDMWriter_Impl2::GribApiCDMWriter_Impl2](#) (const boost::shared_ptr< [CDMReader](#) > & cdmReader, const std::string & outputFile, const std::string & configFile)

12.37.2.2 [virtual MetNoFimex::GribApiCDMWriter_Impl2::~~GribApiCDMWriter_Impl2](#) ()
[virtual]

12.37.3 Member Function Documentation

12.37.3.1 [virtual void MetNoFimex::GribApiCDMWriter_Impl2::setParameter](#) (const std::string & varName, const [FimexTime](#) & fTime, double levelValue) throw (CDMException)
[virtual]

Implements [MetNoFimex::GribApiCDMWriter_ImplAbstract](#).

12.37.3.2 [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](#).

12.37.3.3 virtual void MetNoFimex::GribApiCDMWriter_Impl2::setLevel (const std::string & *varName*, double *levelValue*) [virtual]

Implements [MetNoFimex::GribApiCDMWriter_ImplAbstract](#).

12.37.3.4 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](#).

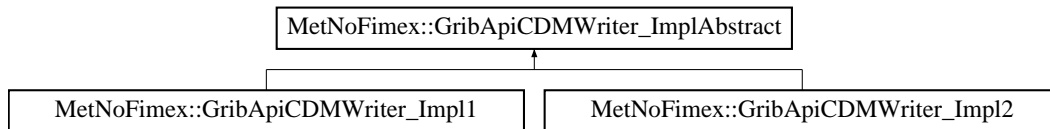
The documentation for this class was generated from the following file:

- include/fimex/[GribApiCDMWriter_Impl2.h](#)

12.38 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 boost::shared_ptr< [CDMReader](#) > [cdmReader](#)
- const **std::string** [outputFile](#)
- const **std::string** [configFile](#)
- const boost::shared_ptr< [XMLDoc](#) > [xmlConfig](#)
- boost::shared_ptr< [grib_handle](#) > [gribHandle](#)
- [LoggerPtr](#) [logger](#)

12.38.1 Constructor & Destructor Documentation

12.38.1.1 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

12.38.1.2 virtual MetNoFimex::GribApiCDMWriter_ImplAbstract::~~GribApiCDMWriter_ImplAbstract () [virtual]

12.38.2 Member Function Documentation

12.38.2.1 void MetNoFimex::GribApiCDMWriter_ImplAbstract::run () throw (CDMException)

actually write the data

The run function has to be called after construction of the object to actually fetch and write the data.

12.38.2.2 virtual void MetNoFimex::GribApiCDMWriter_ImplAbstract::setGlobalAttributes () [protected, virtual]

add the global attributes from the config to the default grib-handle

12.38.2.3 virtual void MetNoFimex::GribApiCDMWriter_ImplAbstract::setData (const boost::shared_ptr< Data > & *data*) [protected, virtual]

12.38.2.4 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](#).

12.38.2.5 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](#).

12.38.2.6 virtual void MetNoFimex::GribApiCDMWriter_ImplAbstract::setTime (const std::string & *varName*, const FimexTime & *fTime*) [protected, virtual]

12.38.2.7 virtual void MetNoFimex::GribApiCDMWriter_ImplAbstract::setLevel (const std::string & *varName*, double *levelValue*) [protected, pure virtual]

Implemented in [MetNoFimex::GribApiCDMWriter_Impl1](#), and [MetNoFimex::GribApiCDMWriter_Impl2](#).

12.38.2.8 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

12.38.2.9 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

12.38.2.10 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](#).

12.38.2.11 virtual void MetNoFimex::GribApiCDMWriter_ImplAbstract::writeGribHandleToFile () [protected, virtual]

12.38.2.12 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

12.38.3 Member Data Documentation

12.38.3.1 `int MetNoFimex::GribApiCDMWriter_ImplAbstract::gribVersion` [protected]

12.38.3.2 `const boost::shared_ptr<CDMReader> MetNoFimex::GribApiCDMWriter_ImplAbstract::cdmReader` [protected]

12.38.3.3 `const std::string MetNoFimex::GribApiCDMWriter_ImplAbstract::outputFile` [protected]

12.38.3.4 `const std::string MetNoFimex::GribApiCDMWriter_ImplAbstract::configFile` [protected]

12.38.3.5 `const boost::shared_ptr<XMLDoc> MetNoFimex::GribApiCDMWriter_ImplAbstract::xmlConfig` [protected]

12.38.3.6 `boost::shared_ptr<grib_handle> MetNoFimex::GribApiCDMWriter_ImplAbstract::gribHandle` [protected]

12.38.3.7 `LoggerPtr MetNoFimex::GribApiCDMWriter_ImplAbstract::logger` [protected]

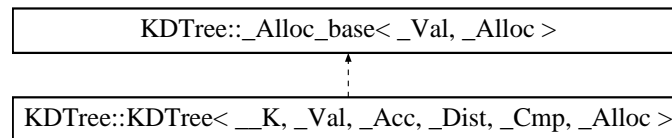
The documentation for this class was generated from the following file:

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

12.39 `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >` Class Template Reference

```
#include <kdtree.hpp>
```

Inheritance diagram for `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::`



Public Types

- typedef `_Region< __K, _Val, typename _Acc::result_type, _Acc, _Cmp >` `_Region_`
- typedef `_Val` `value_type`
- typedef `value_type * pointer`
- typedef `value_type const * const_pointer`
- typedef `value_type & reference`
- typedef `value_type const & const_reference`
- typedef `_Acc::result_type` `subvalue_type`
- typedef `_Dist::distance_type` `distance_type`
- typedef `size_t` `size_type`
- typedef `ptrdiff_t` `difference_type`
- typedef `_Iterator< _Val, const_reference, const_pointer >` `const_iterator`
- typedef `const_iterator` `iterator`
- typedef `std::reverse_iterator< const_iterator >` `const_reverse_iterator`
- typedef `std::reverse_iterator< iterator >` `reverse_iterator`

Public Member Functions

- `KDTree` (`_Acc` const & `_acc= _Acc()`, `_Dist` const & `_dist= _Dist()`, `_Cmp` const & `_cmp= _Cmp()`, const `allocator_type` & `_a=allocator_type()`)
- `KDTree` (const `KDTree` & `_x`)
- `template<typename _InputIterator>`
`KDTree` (`_InputIterator` `__first`, `_InputIterator` `__last`, `_Acc` const & `acc= _Acc()`, `_Dist` const & `__dist= _Dist()`, `_Cmp` const & `__cmp= _Cmp()`, const `allocator_type` & `_a=allocator_type()`)
- void `efficient_replace_and_optimise` (`std::vector< value_type >` & `writable_vector`)
- `KDTree` & `operator=` (const `KDTree` & `_x`)
- `~KDTree` ()
- `allocator_type` `get_allocator` () const
- `size_type` `size` () const
- `size_type` `max_size` () const
- bool `empty` () const
- void `clear` ()
- `_Cmp` `value_comp` () const

Comparator for the values in the `KDTree`.

- [_Acc value_acc](#) () const
Accessor to the value's elements.
- [const _Dist & value_distance](#) () const
Distance calculator between 2 value's element.
- [_Dist & value_distance](#) ()
- [const_iterator begin](#) () const
- [const_iterator end](#) () const
- [const_reverse_iterator rbegin](#) () const
- [const_reverse_iterator rend](#) () const
- [iterator insert](#) (iterator, [const_reference](#) __V)
- [iterator insert](#) ([const_reference](#) __V)
- [template<class _InputIterator>](#)
[void insert](#) (_InputIterator __first, _InputIterator __last)
- [void insert](#) (iterator __pos, [size_type](#) __n, [const value_type](#) &__x)
- [template<typename _InputIterator>](#)
[void insert](#) (iterator __pos, _InputIterator __first, _InputIterator __last)
- [void erase](#) ([const_reference](#) __V)
- [void erase_exact](#) ([const_reference](#) __V)
- [void erase](#) ([const_iterator](#) const &__IT)
- [template<class SearchVal>](#)
[const_iterator find](#) (SearchVal const &__V) const
- [template<class SearchVal>](#)
[const_iterator find_exact](#) (SearchVal const &__V) const
- [size_type count_within_range](#) ([const_reference](#) __V, [subvalue_type](#) const __R) const
- [size_type count_within_range](#) (_Region_ const &__REGION) const
- [template<typename SearchVal, class Visitor>](#)
[Visitor visit_within_range](#) (SearchVal const &V, [subvalue_type](#) const R, Visitor visitor) const
- [template<class Visitor>](#)
[Visitor visit_within_range](#) (_Region_ const ®ION, Visitor visitor) const
- [const_iterator find_within_range_iterative](#) ([const_reference](#) __a, [const_reference](#) __b)
- [template<typename SearchVal, typename _OutputIterator>](#)
[_OutputIterator find_within_range](#) (SearchVal const &val, [subvalue_type](#) const range, _-
OutputIterator out) const
- [template<typename _OutputIterator>](#)
[_OutputIterator find_within_range](#) (_Region_ const ®ion, _OutputIterator out) const
- [template<class SearchVal>](#)
std::pair< [const_iterator](#), [distance_type](#) > [find_nearest](#) (SearchVal const &__val) const
- [template<class SearchVal>](#)
std::pair< [const_iterator](#), [distance_type](#) > [find_nearest](#) (SearchVal const &__val, [distance_type](#) _-
__max) const
- [template<class SearchVal, class _Predicate>](#)
std::pair< [const_iterator](#), [distance_type](#) > [find_nearest_if](#) (SearchVal const &__val, [distance_type](#)
__max, _Predicate __p) const
- [void optimise](#) ()
- [void optimize](#) ()
- [void check_tree](#) ()

Protected Types

- typedef [_Alloc_base](#)< [_Val](#), [_Alloc](#) > [_Base](#)
- typedef [_Base::allocator_type](#) [allocator_type](#)
- typedef [_Node_base](#) * [_Base_ptr](#)
- typedef [_Node_base](#) const * [_Base_const_ptr](#)
- typedef [_Node](#)< [_Val](#) > * [_Link_type](#)
- typedef [_Node](#)< [_Val](#) > const * [_Link_const_type](#)
- typedef [_Node_compare](#)< [_Val](#), [_Acc](#), [_Cmp](#) > [_Node_compare_](#)

Protected Member Functions

- void [_M_check_children](#) ([_Link_const_type](#) child, [_Link_const_type](#) parent, [size_type](#) const level, bool to_the_left)
- void [_M_check_node](#) ([_Link_const_type](#) node, [size_type](#) const level)
- void [_M_empty_initialise](#) ()
- iterator [_M_insert_left](#) ([_Link_type](#) __N, const_reference __V)
- iterator [_M_insert_right](#) ([_Link_type](#) __N, const_reference __V)
- iterator [_M_insert](#) ([_Link_type](#) __N, const_reference __V, [size_type](#) const __L)
- [_Link_type](#) [_M_erase](#) ([_Link_type](#) dead_dad, [size_type](#) const level)
- [_Link_type](#) [_M_get_erase_replacement](#) ([_Link_type](#) node, [size_type](#) const level)
- **std::pair**< [_Link_type](#), [size_type](#) > [_M_get_j_min](#) (**std::pair**< [_Link_type](#), [size_type](#) > const node, [size_type](#) const level)
- **std::pair**< [_Link_type](#), [size_type](#) > [_M_get_j_max](#) (**std::pair**< [_Link_type](#), [size_type](#) > const node, [size_type](#) const level)
- void [_M_erase_subtree](#) ([_Link_type](#) __n)
- const_iterator [_M_find](#) ([_Link_const_type](#) node, const_reference value, [size_type](#) const level) const
- const_iterator [_M_find_exact](#) ([_Link_const_type](#) node, const_reference value, [size_type](#) const level) const
- bool [_M_matches_node_in_d](#) ([_Link_const_type](#) __N, const_reference __V, [size_type](#) const __L) const
- bool [_M_matches_node_in_other_ds](#) ([_Link_const_type](#) __N, const_reference __V, [size_type](#) const __L=0) const
- bool [_M_matches_node](#) ([_Link_const_type](#) __N, const_reference __V, [size_type](#) __L=0) const
- [size_type](#) [_M_count_within_range](#) ([_Link_const_type](#) __N, [_Region](#) const &__REGION, [_Region](#) const &__BOUNDS, [size_type](#) const __L) const
- template<class Visitor>
Visitor [_M_visit_within_range](#) (Visitor visitor, [_Link_const_type](#) N, [_Region](#) const ®ION, [_Region](#) const &BOUNDS, [size_type](#) const L) const
- template<typename _OutputIterator>
[_OutputIterator](#) [_M_find_within_range](#) ([_OutputIterator](#) out, [_Link_const_type](#) __N, [_Region](#) const &__REGION, [_Region](#) const &__BOUNDS, [size_type](#) const __L) const
- template<typename _Iter>
void [_M_optimise](#) ([_Iter](#) const &__A, [_Iter](#) const &__B, [size_type](#) const __L)
- [_Link_const_type](#) [_M_get_root](#) () const
- [_Link_type](#) [_M_get_root](#) ()
- void [_M_set_root](#) ([_Link_type](#) n)
- [_Link_const_type](#) [_M_get_leftmost](#) () const
- void [_M_set_leftmost](#) ([_Node_base](#) *a)
- [_Link_const_type](#) [_M_get_rightmost](#) () const
- void [_M_set_rightmost](#) ([_Node_base](#) *a)

- [_Link_type _M_new_node](#) (const_reference __V, [_Base_ptr](#) const __PARENT=NULL, [_Base_ptr](#) const __LEFT=NULL, [_Base_ptr](#) const __RIGHT=NULL)
- void [_M_delete_node](#) ([_Link_type](#) __p)

Static Protected Member Functions

- static [_Link_type _S_parent](#) ([_Base_ptr](#) N)
- static [_Link_const_type _S_parent](#) ([_Base_const_ptr](#) N)
- static void [_S_set_parent](#) ([_Base_ptr](#) N, [_Base_ptr](#) p)
- static void [_S_set_left](#) ([_Base_ptr](#) N, [_Base_ptr](#) l)
- static [_Link_type _S_left](#) ([_Base_ptr](#) N)
- static [_Link_const_type _S_left](#) ([_Base_const_ptr](#) N)
- static void [_S_set_right](#) ([_Base_ptr](#) N, [_Base_ptr](#) r)
- static [_Link_type _S_right](#) ([_Base_ptr](#) N)
- static [_Link_const_type _S_right](#) ([_Base_const_ptr](#) N)
- static bool [_S_is_leaf](#) ([_Base_const_ptr](#) N)
- static const_reference [_S_value](#) ([_Link_const_type](#) N)
- static const_reference [_S_value](#) ([_Base_const_ptr](#) N)
- static [_Link_const_type _S_minimum](#) ([_Link_const_type](#) __X)
- static [_Link_const_type _S_maximum](#) ([_Link_const_type](#) __X)

Protected Attributes

- [_Link_type _M_root](#)
- [_Node_base _M_header](#)
- [size_type _M_count](#)
- [_Acc _M_acc](#)
- [_Cmp _M_cmp](#)
- [_Dist _M_dist](#)

```
template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename
_Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename
_Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>
>> class KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >
```

12.39.1 Member Typedef Documentation

12.39.1.1 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> typedef _Alloc_base<_Val, _Alloc> KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_Base` [protected]

12.39.1.2 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> typedef _Base::allocator_type KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_allocator_type` [protected]

Reimplemented from [KDTree::_Alloc_base<_Val, _Alloc >](#).

12.39.1.3 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> typedef _Node_base* KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_Base_ptr` [protected]

Reimplemented from [KDTree::_Alloc_base<_Val, _Alloc >](#).

- 12.39.1.4 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> typedef _Node_base const* KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::Base_const_ptr`
[protected]
- 12.39.1.5 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> typedef _Node<_Val>* KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::Link_type`
[protected]
- 12.39.1.6 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> typedef _Node<_Val> const* KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::Link_const_type`
[protected]
- 12.39.1.7 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> typedef _Node_compare<_Val, _Acc, _Cmp> KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::Node_compare_` [protected]
- 12.39.1.8 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> typedef _Region<__K, _Val, typename _Acc::result_type, _Acc, _Cmp> KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::Region_`
- 12.39.1.9 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> typedef _Val KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::value_type`
- 12.39.1.10 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> typedef value_type* KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::pointer`
- 12.39.1.11 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> typedef value_type const* KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::const_pointer`
- 12.39.1.12 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> typedef value_type& KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::reference`
- 12.39.1.13 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>`

12.39.2.2 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::KDTree (const KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc > & __x) [inline]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_empty_initialise(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_optimise(), std::back_inserter(), std::vector< _Tp, _Alloc >::begin(), std::copy(), std::vector< _Tp, _Alloc >::end(), and std::vector< _Tp, _Alloc >::reserve().

12.39.2.3 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> template<typename _InputIterator> KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::KDTree (_InputIterator __first, _InputIterator __last, _Acc const & acc = _Acc(), _Dist const & __dist = _Dist(), _Cmp const & __cmp = _Cmp(), const allocator_type & __a = allocator_type()) [inline]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_empty_initialise(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_optimise(), std::back_inserter(), std::vector< _Tp, _Alloc >::begin(), std::copy(), std::distance(), std::vector< _Tp, _Alloc >::end(), and std::vector< _Tp, _Alloc >::reserve().

12.39.2.4 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::~KDTree () [inline]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::clear().

12.39.3 Member Function Documentation

12.39.3.1 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> void KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::efficient_replace_and_optimise (std::vector< value_type > & writable_vector) [inline]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_optimise(), std::vector< _Tp, _Alloc >::begin(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::clear(), and std::vector< _Tp, _Alloc >::end().

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::operator=().

12.39.3.2 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>,
typename _Dist = squared_difference<typename _Acc::result_type, typename
_Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>,
typename _Alloc = std::allocator<_Node<_Val> >> KDTree& KDTree::KDTree<
__K, _Val, _Acc, _Dist, _Cmp, _Alloc >::operator=(const KDTree< __K, _Val, _Acc,
_Dist, _Cmp, _Alloc > & __x) [inline]`

References `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_acc`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_cmp`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_dist`, `std::back_inserter()`, `std::copy()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::efficient_replace_and_optimise()`, and `std::vector< _Tp, _Alloc >::reserve()`.

12.39.3.3 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>,
typename _Dist = squared_difference<typename _Acc::result_type, typename
_Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>,
typename _Alloc = std::allocator<_Node<_Val> >> allocator_type KDTree::KDTree<
__K, _Val, _Acc, _Dist, _Cmp, _Alloc >::get_allocator () const [inline]`

Reimplemented from `KDTree::_Alloc_base< _Val, _Alloc >`.

References `KDTree::_Alloc_base< _Val, _Alloc >::get_allocator()`.

12.39.3.4 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>,
typename _Dist = squared_difference<typename _Acc::result_type, typename
_Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>,
typename _Alloc = std::allocator<_Node<_Val> >> size_type KDTree::KDTree< __K,
_Val, _Acc, _Dist, _Cmp, _Alloc >::size () const [inline]`

References `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_count`.

Referenced by `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::empty()`.

12.39.3.5 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>,
typename _Dist = squared_difference<typename _Acc::result_type, typename
_Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>,
typename _Alloc = std::allocator<_Node<_Val> >> size_type KDTree::KDTree< __K,
_Val, _Acc, _Dist, _Cmp, _Alloc >::max_size () const [inline]`

12.39.3.6 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>,
typename _Dist = squared_difference<typename _Acc::result_type, typename
_Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>,
typename _Alloc = std::allocator<_Node<_Val> >> bool KDTree::KDTree< __K,
_Val, _Acc, _Dist, _Cmp, _Alloc >::empty () const [inline]`

References `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::size()`.

12.39.3.7 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> void KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::clear () [inline]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_count, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_erase_subtree(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_root(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_header, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_set_leftmost(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_set_rightmost(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_set_root().

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::efficient_replace_and_optimise(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::optimise(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::~~KDTree().

12.39.3.8 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> _Cmp KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::value_comp () const [inline]`

Comparator for the values in the [KDTree](#).

The comparator shall not be modified, it could invalidate the tree.

Returns:

a copy of the comparator used by the [KDTree](#).

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_cmp.

12.39.3.9 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> _Acc KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::value_acc () const [inline]`

Accessor to the value's elements.

This accessor shall not be modified, it could invalidate the tree.

Returns:

a copy of the accessor used by the [KDTree](#).

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_acc.

12.39.3.10 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> const _Dist& KDTree::KDTree<__K, _Val, _Acc, _Dist, _Cmp, _Alloc>::value_distance () const [inline]`

Distance calculator between 2 value's element.

This functor can be modified. It's modification will only affect the behavior of the find and find_nearest functions.

Returns:

a reference to the distance calculator used by the [KDTree](#).

References `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_dist`.

12.39.3.11 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> _Dist& KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::value_distance () [inline]`

References `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_dist`.

12.39.3.12 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> const_iterator KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::begin () const [inline]`

References `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_leftmost()`.

Referenced by `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_within_range_iterative()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::optimise()`, and `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::rend()`.

12.39.3.13 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> const_iterator KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::end () const [inline]`

References `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_header`.

Referenced by `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_find()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_find_exact()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::erase()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_exact()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_nearest()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_nearest_if()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::optimise()`, and `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::rbegin()`.

12.39.3.14 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> const_reverse_iterator KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::rbegin() const [inline]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::end().

12.39.3.15 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> const_reverse_iterator KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::rend() const [inline]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::begin().

12.39.3.16 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> iterator KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::insert(iterator, const_reference __V) [inline]`

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_optimise(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::insert().

12.39.3.17 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> iterator KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::insert(const_reference __V) [inline]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_count, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_get_root(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_header, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_insert(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_new_node(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_set_leftmost(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_set_rightmost(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_set_root().

12.39.3.18 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> template<class InputIterator> void KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::insert(InputIterator __first, InputIterator __last) [inline]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::insert().

12.39.3.19 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> void KDTree::KDTree<__K, _Val, _Acc, _Dist, _Cmp, _Alloc>::insert(iterator __pos, size_type __n, const value_type & __x) [inline]`

References `KDTree::KDTree<__K, _Val, _Acc, _Dist, _Cmp, _Alloc>::insert()`.

12.39.3.20 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> template<typename _InputIterator> void KDTree::KDTree<__K, _Val, _Acc, _Dist, _Cmp, _Alloc>::insert(iterator __pos, _InputIterator __first, _InputIterator __last) [inline]`

References `KDTree::KDTree<__K, _Val, _Acc, _Dist, _Cmp, _Alloc>::insert()`.

12.39.3.21 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> void KDTree::KDTree<__K, _Val, _Acc, _Dist, _Cmp, _Alloc>::erase(const_reference __V) [inline]`

References `KDTree::KDTree<__K, _Val, _Acc, _Dist, _Cmp, _Alloc>::find()`.

Referenced by `KDTree::KDTree<__K, _Val, _Acc, _Dist, _Cmp, _Alloc>::erase_exact()`.

12.39.3.22 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> void KDTree::KDTree<__K, _Val, _Acc, _Dist, _Cmp, _Alloc>::erase_exact(const_reference __V) [inline]`

References `KDTree::KDTree<__K, _Val, _Acc, _Dist, _Cmp, _Alloc>::erase()`, and `KDTree::KDTree<__K, _Val, _Acc, _Dist, _Cmp, _Alloc>::find_exact()`.

12.39.3.23 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> void KDTree::KDTree<__K, _Val, _Acc, _Dist, _Cmp, _Alloc>::erase(const_iterator const & __IT) [inline]`

References `KDTree::KDTree<__K, _Val, _Acc, _Dist, _Cmp, _Alloc>::M_count`, `KDTree::KDTree<__K, _Val, _Acc, _Dist, _Cmp, _Alloc>::M_delete_node()`, `KDTree::KDTree<__K, _Val, _Acc, _Dist, _Cmp, _Alloc>::M_erase()`, `KDTree::KDTree<__K, _Val, _Acc, _Dist, _Cmp, _Alloc>::M_header`, `KDTree::KDTree<__K, _Val, _Acc, _Dist, _Cmp, _Alloc>::S_parent()`, `KDTree::KDTree<__K, _Val, _Acc, _Dist, _Cmp, _Alloc>::end()`, and `KDTree::Iterator<_Val, _Ref, _Ptr>::get_raw_node()`.

12.39.3.24 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> template<class SearchVal> const_iterator KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find (SearchVal const & __V) const [inline]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_find(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_root(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::end().

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::erase().

12.39.3.25 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> template<class SearchVal> const_iterator KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_exact (SearchVal const & __V) const [inline]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_find_exact(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_root(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::end().

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::erase_exact().

12.39.3.26 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> size_type KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::count_within_range (const_reference __V, subvalue_type const __R) const [inline]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_acc, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_cmp, and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_root().

12.39.3.27 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> size_type KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::count_within_range (_Region const & __REGION) const [inline]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_count_within_range(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_root().

12.39.3.28 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> template<typename SearchVal, class Visitor> Visitor KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::visit_within_range (SearchVal const & V, subvalue_type const R, Visitor visitor) const [inline]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_acc, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_cmp, and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_root().

12.39.3.29 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> template<class Visitor> Visitor KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::visit_within_range (_Region_const & REGION, Visitor visitor) const [inline]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_root(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_visit_within_range().

12.39.3.30 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> const_iterator KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_within_range_iterative (const_reference __a, const_reference __b) [inline]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::begin().

12.39.3.31 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> template<typename SearchVal, typename _OutputIterator> _OutputIterator KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_within_range (SearchVal const & val, subvalue_type const range, _OutputIterator out) const [inline]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_acc, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_cmp, and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_root().

12.39.3.32 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> template<typename _OutputIterator> _OutputIterator KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_within_range (_Region_ const & region, _OutputIterator out) const [inline]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_find_within_range(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_root().

12.39.3.33 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> template<class SearchVal> std::pair<const_iterator, distance_type> KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_nearest (SearchVal const & __val) const [inline]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_acc, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_cmp, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_dist, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_root(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_header, KDTree::_S_accumulate_node_distance(), KDTree::_S_node_nearest(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::end(), std::pair<_T1, _T2 >::first, and std::pair<_T1, _T2 >::second.

12.39.3.34 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> template<class SearchVal> std::pair<const_iterator, distance_type> KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_nearest (SearchVal const & __val, distance_type __max) const [inline]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_acc, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_cmp, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_dist, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_root(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_header, KDTree::_S_accumulate_node_distance(), KDTree::_S_node_nearest(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::end(), std::pair<_T1, _T2 >::first, and std::pair<_T1, _T2 >::second.

12.39.3.35 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> template<class SearchVal, class _Predicate> std::pair<const_iterator, distance_type> KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_nearest_if (SearchVal const & __val, distance_type __max, _Predicate __p) const [inline]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_acc, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_cmp, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_dist, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_root(),

KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_header, KDTree::_S_accumulate_node_distance(), KDTree::_S_node_nearest(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::end(), std::pair< _T1, _T2 >::first, and std::pair< _T1, _T2 >::second.

12.39.3.36 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> void KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::optimise () [inline]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_optimise(), std::vector< _Tp, _Alloc >::begin(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::begin(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::clear(), std::vector< _Tp, _Alloc >::end(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::end().

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::optimize().

12.39.3.37 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> void KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::optimize () [inline]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::optimise().

12.39.3.38 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> void KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::check_tree () [inline]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_check_node(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_root().

12.39.3.39 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> void KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_check_children (_Link_const_type child, _Link_const_type parent, size_type const level, bool to_the_left) [inline, protected]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_acc, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_cmp, KDTree::_Node< _Val >::_M_value, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_left(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_right().

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_check_node().

12.39.3.40 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> void KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_check_node (Link_const_type node, size_type const level) [inline, protected]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_check_children(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_left(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_right().

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::check_tree().

12.39.3.41 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> void KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_empty_initialise () [inline, protected]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_header, KDTree::_Node_base::_M_parent, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_set_leftmost(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_set_rightmost(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_set_root().

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::KDTree().

12.39.3.42 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> iterator KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_insert_left (Link_type __N, const_reference __V) [inline, protected]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_count, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_leftmost(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_new_node(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_set_leftmost(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_left(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_set_left(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_set_parent().

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_insert().

12.39.3.43 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> iterator KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_insert_right (Link_type __N, const_reference __V) [inline, protected]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_count, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_rightmost(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_new_node(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_set_rightmost(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_right(),

KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_set_parent(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_set_right().

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_insert().

12.39.3.44 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> iterator KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_insert (_Link_type __N, const_reference __V, size_type const __L) [inline, protected]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_acc, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_cmp, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_rightmost(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_insert_left(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_insert_right(), KDTree::_Node< _Val >::_M_value, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_left(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_right().

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::insert().

12.39.3.45 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> _Link_type KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_erase (_Link_type dead_dad, size_type const level) [inline, protected]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_erase_replacement(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_leftmost(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_rightmost(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_root(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_set_leftmost(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_set_rightmost(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_set_root(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_left(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_parent(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_right(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_set_left(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_set_parent(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_set_right().

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_erase_replacement(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::erase().

12.39.3.46 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> _Link_type KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_erase_replacement (_Link_type node, size_type const level) [inline, protected]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_acc, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_cmp, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_erase(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_j_max(),

KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_j_min(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_is_leaf(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_left(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_parent(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_right(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_set_left(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_set_right(), std::pair< _T1, _T2 >::first, and std::pair< _T1, _T2 >::second.

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_erase().

12.39.3.47 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> std::pair<_Link_type, size_type>> KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_j_min (std::pair<_Link_type, size_type> const node, size_type const level) [inline, protected]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_acc, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_cmp, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_is_leaf(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_left(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_right(), std::pair< _T1, _T2 >::first, left(), right(), and std::pair< _T1, _T2 >::second.

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_erase_replacement().

12.39.3.48 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> std::pair<_Link_type, size_type>> KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_j_max (std::pair<_Link_type, size_type> const node, size_type const level) [inline, protected]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_acc, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_cmp, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_is_leaf(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_left(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_right(), std::pair< _T1, _T2 >::first, left(), right(), and std::pair< _T1, _T2 >::second.

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_erase_replacement().

12.39.3.49 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> void KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_erase_subtree (_Link_type __n) [inline, protected]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_delete_node(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_left(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_right().

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::clear().

12.39.3.50 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> const_iterator KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_find(_Link_const_type node, const_reference value, size_type const level) const` `[inline, protected]`

References `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_acc`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_cmp`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_matches_node()`, `KDTree::Node< _Val >::M_value`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::S_left()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::S_right()`, and `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::end()`.

Referenced by `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find()`.

12.39.3.51 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> const_iterator KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_find_exact(_Link_const_type node, const_reference value, size_type const level) const` `[inline, protected]`

References `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_acc`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_cmp`, `KDTree::Node< _Val >::M_value`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::S_left()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::S_right()`, and `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::end()`.

Referenced by `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_exact()`.

12.39.3.52 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> bool KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_matches_node_in_d(_Link_const_type _N, const_reference _V, size_type const _L) const` `[inline, protected]`

References `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_acc`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_cmp`, and `KDTree::Node< _Val >::M_value`.

Referenced by `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_matches_node()`, and `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_matches_node_in_other_ds()`.

12.39.3.53 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> bool KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_matches_node_in_other_ds(_Link_const_type _N, const_reference _V, size_type const _L = 0) const` `[inline, protected]`

References `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_matches_node_in_d()`.

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_matches_node().

12.39.3.54 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> bool KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_matches_node (_Link_const_type __N, const_reference __V, size_type __L = 0) const [inline, protected]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_matches_node_in_d(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_matches_node_in_other_ds().

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_find().

12.39.3.55 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> size_type KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_count_within_range (_Link_const_type __N, _Region_const & __REGION, _Region_const & __BOUNDS, size_type const __L) const [inline, protected]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_left(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_right(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_value(), count(), KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::encloses(), KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::intersects_with(), KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::set_high_bound(), and KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::set_low_bound().

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::count_within_range().

12.39.3.56 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> template<class Visitor> Visitor KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_visit_within_range (Visitor visitor, _Link_const_type N, _Region_const & __REGION, _Region_const & __BOUNDS, size_type const L) const [inline, protected]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_left(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_right(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_value(), KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::encloses(), KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::intersects_with(), KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::set_high_bound(), and KDTree::_Region< __K, _Val, _SubVal, _Acc, _Cmp >::set_low_bound().

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::visit_within_range().

12.39.3.57 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> template<typename _OutputIterator> _OutputIterator KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_find_within_range (_OutputIterator out, _Link_const_type __N, _Region_const & __REGION, _Region_const & __BOUNDS, size_type const __L) const [inline, protected]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::S_left(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::S_right(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::S_value(), KDTree::Region< __K, _Val, _SubVal, _Acc, _Cmp >::encloses(), KDTree::Region< __K, _Val, _SubVal, _Acc, _Cmp >::intersects_with(), KDTree::Region< __K, _Val, _SubVal, _Acc, _Cmp >::set_high_bound(), and KDTree::Region< __K, _Val, _SubVal, _Acc, _Cmp >::set_low_bound().

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_within_range().

12.39.3.58 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> template<typename _Iter> void KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_optimise (_Iter const & __A, _Iter const & __B, size_type const __L) [inline, protected]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_acc, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_cmp, KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::insert(), and std::nth_element().

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::efficient_replace_and_optimise(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::KDTree(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::optimise().

12.39.3.59 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> _Link_const_type KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_get_root () const [inline, protected]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_root.

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_erase(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::check_tree(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::clear(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::count_within_range(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_exact(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_nearest(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_nearest_if(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_within_range(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::insert(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::visit_within_range().

12.39.3.60 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> _Link_type KDTree::KDTree<__K, _Val, _Acc, _Dist, _Cmp, _Alloc>::M_get_root () [inline, protected]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_root.

12.39.3.61 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> void KDTree::KDTree<__K, _Val, _Acc, _Dist, _Cmp, _Alloc>::M_set_root (_Link_type n) [inline, protected]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_root.

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_empty_initialise(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_erase(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::clear(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::insert().

12.39.3.62 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> _Link_const_type KDTree::KDTree<__K, _Val, _Acc, _Dist, _Cmp, _Alloc>::M_get_leftmost () const [inline, protected]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_header, and KDTree::_Node_base::M_left.

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_erase(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_insert_left(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::begin().

12.39.3.63 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> void KDTree::KDTree<__K, _Val, _Acc, _Dist, _Cmp, _Alloc>::M_set_leftmost (_Node_base * a) [inline, protected]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_header, and KDTree::_Node_base::M_left.

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_empty_initialise(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_erase(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_insert_left(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::clear(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::insert().

12.39.3.64 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> _Link_const_type
 KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_get_rightmost ()
 const [inline, protected]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_header, and KDTree::_Node_base::M_right.

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_erase(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_insert(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_insert_right().

12.39.3.65 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> void KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_set_rightmost (_Node_base *a) [inline, protected]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_header, and KDTree::_Node_base::M_right.

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_empty_initialise(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_erase(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_insert_right(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::clear(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::insert().

12.39.3.66 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> static _Link_type
 KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::S_parent (_Base_ptr N)
 [inline, static, protected]`

References KDTree::_Node_base::M_parent.

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_erase(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::M_erase_replacement(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::erase().

12.39.3.67 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> static _Link_const_type KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_parent (_Base_const_ptr N) [inline, static, protected]`

12.39.3.68 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> static void KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_set_parent (_Base_ptr N, _Base_ptr p) [inline, static, protected]`

References KDTree::_Node_base::_M_parent.

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_erase(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_insert_left(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_insert_right().

12.39.3.69 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> static void KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_set_left (_Base_ptr N, _Base_ptr l) [inline, static, protected]`

References KDTree::_Node_base::_M_left.

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_erase(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_erase_replacement(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_insert_left().

12.39.3.70 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> static _Link_type KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_left (_Base_ptr N) [inline, static, protected]`

References KDTree::_Node_base::_M_left.

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_check_children(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_check_node(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_count_within_range(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_erase(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_erase_subtree(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_find(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_find_exact(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_find_within_range(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_erase_replacement(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_j_max(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_j_min(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_insert(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_insert_left(), KDTree::KDTree< __K, _Val, _Acc, _Dist,

`_Cmp, _Alloc >::_M_visit_within_range()`, and `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_is_leaf()`.

12.39.3.71 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> static _Link_const_type KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_left (_Base_const_ptr N)` [`inline, static, protected`]

12.39.3.72 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> static void KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_set_right (_Base_ptr N, _Base_ptr r)` [`inline, static, protected`]

References `KDTree::_Node_base::_M_right`.

Referenced by `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_erase()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_erase_replacement()`, and `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_insert_right()`.

12.39.3.73 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> static _Link_type KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_right (_Base_ptr N)` [`inline, static, protected`]

References `KDTree::_Node_base::_M_right`.

Referenced by `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_check_children()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_check_node()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_count_within_range()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_erase()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_erase_subtree()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_find()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_find_exact()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_find_within_range()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_erase_replacement()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_j_max()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_j_min()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_insert()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_insert_right()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_visit_within_range()`, and `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_is_leaf()`.

12.39.3.74 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> static _Link_const_type KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_right (_Base_const_ptr N) [inline, static, protected]`

12.39.3.75 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> static bool KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_is_leaf (_Base_const_ptr N) [inline, static, protected]`

References KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_left(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_right().

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_erase_replacement(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_j_max(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_j_min().

12.39.3.76 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> static const_reference KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_value (_Link_const_type N) [inline, static, protected]`

References KDTree::_Node< _Val >::_M_value.

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_count_within_range(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_find_within_range(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_visit_within_range().

12.39.3.77 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> static const_reference KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_value (_Base_const_ptr N) [inline, static, protected]`

12.39.3.78 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val> >> static _Link_const_type KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_minimum (_Link_const_type _X) [inline, static, protected]`

References KDTree::_Node_base::_S_minimum().

12.39.3.79 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> static _Link_const_type KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_S_maximum (_Link_const_type __X) [inline, static, protected]`

References KDTree::_Node_base::_S_maximum().

12.39.3.80 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> _Link_type KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_new_node (const_reference __V, _Base_ptr const __PARENT = NULL, _Base_ptr const __LEFT = NULL, _Base_ptr const __RIGHT = NULL) [inline, protected]`

References KDTree::_Alloc_base<_Val, _Alloc >::_M_construct_node().

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_insert_left(), KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_insert_right(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::insert().

12.39.3.81 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> void KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_delete_node (_Link_type __p) [inline, protected]`

References KDTree::_Alloc_base<_Val, _Alloc >::_M_deallocate_node(), and KDTree::_Alloc_base<_Val, _Alloc >::_M_destroy_node().

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_erase_subtree(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::erase().

12.39.4 Member Data Documentation

12.39.4.1 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>, typename _Dist = squared_difference<typename _Acc::result_type, typename _Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>, typename _Alloc = std::allocator<_Node<_Val>>> _Link_type KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_root [protected]`

Referenced by KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_root(), and KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_set_root().

12.39.4.2 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>,
typename _Dist = squared_difference<typename _Acc::result_type, typename
_Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>,
typename _Alloc = std::allocator<_Node<_Val> >> _Node_base KDTree::KDTree<
__K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_header` [protected]

Referenced by `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_empty_initialise()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_leftmost()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_rightmost()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_set_leftmost()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_set_rightmost()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::clear()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::end()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::erase()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_nearest()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_nearest_if()`, and `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::insert()`.

12.39.4.3 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>,
typename _Dist = squared_difference<typename _Acc::result_type, typename
_Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>,
typename _Alloc = std::allocator<_Node<_Val> >> size_type KDTree::KDTree< __K,
_Val, _Acc, _Dist, _Cmp, _Alloc >::_M_count` [protected]

Referenced by `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_insert_left()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_insert_right()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::clear()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::erase()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::insert()`, and `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::size()`.

12.39.4.4 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>,
typename _Dist = squared_difference<typename _Acc::result_type, typename
_Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>,
typename _Alloc = std::allocator<_Node<_Val> >> _Acc KDTree::KDTree< __K,
_Val, _Acc, _Dist, _Cmp, _Alloc >::_M_acc` [protected]

Referenced by `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_check_children()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_find()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_find_exact()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_erase_replacement()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_j_max()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_j_min()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_insert()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_matches_node_in_d()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_optimise()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::count_within_range()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_nearest()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_nearest_if()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_within_range()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::operator=()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::value_acc()`, and `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::visit_within_range()`.

12.39.4.5 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>,
typename _Dist = squared_difference<typename _Acc::result_type, typename
_Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>,
typename _Alloc = std::allocator<_Node<_Val> >> _Cmp KDTree::KDTree< __K,
_Val, _Acc, _Dist, _Cmp, _Alloc >::_M_cmp [protected]`

Referenced by `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_check_children()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_find()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_find_exact()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_erase_replacement()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_j_max()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_get_j_min()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_insert()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_matches_node_in_d()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::_M_optimise()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::count_within_range()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_nearest()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_nearest_if()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_within_range()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::operator=()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::value_comp()`, and `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::visit_within_range()`.

12.39.4.6 `template<size_t const __K, typename _Val, typename _Acc = _Bracket_accessor<_Val>,
typename _Dist = squared_difference<typename _Acc::result_type, typename
_Acc::result_type>, typename _Cmp = std::less<typename _Acc::result_type>,
typename _Alloc = std::allocator<_Node<_Val> >> _Dist KDTree::KDTree< __K,
_Val, _Acc, _Dist, _Cmp, _Alloc >::_M_dist [protected]`

Referenced by `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_nearest()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::find_nearest_if()`, `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::operator=()`, and `KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >::value_distance()`.

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

- [include/kdtree++/kdtree.hpp](#)

12.40 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)

12.40.1 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.

12.40.2 Member Enumeration Documentation

12.40.2.1 enum MetNoFimex::Logger::LogLevel

different log levels

Enumerator:

OFF
FATAL
ERROR
WARN
INFO
DEBUG

12.40.3 Constructor & Destructor Documentation

12.40.3.1 `MetNoFimex::Logger::Logger` (const `std::string` & *className*)

12.40.3.2 virtual `MetNoFimex::Logger::~~Logger` () [virtual]

12.40.4 Member Function Documentation

12.40.4.1 virtual bool `MetNoFimex::Logger::isEnabledFor` (`LogLevel` *level*) [virtual]

check if the loglevel of this logger is active

12.40.4.2 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__`

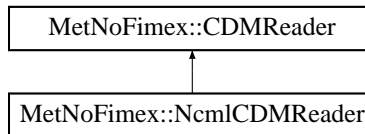
The documentation for this class was generated from the following file:

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

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

12.41.1 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.

12.41.2 Constructor & Destructor Documentation

12.41.2.1 MetNoFimex::NcmlCDMReader::NcmlCDMReader (**std::string** configFile) throw (CDMException)

Parameters:

configFile ncml-file with location set

Exceptions:

[CDMException](#)

12.41.2.2 `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](#)

12.41.2.3 `virtual MetNoFimex::NcmlCDMReader::~~NcmlCDMReader () [virtual]`

12.41.3 Member Function Documentation

12.41.3.1 `virtual const 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](#).

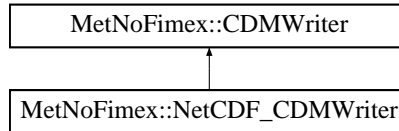
The documentation for this class was generated from the following file:

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

12.42 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)

12.42.1 Constructor & Destructor Documentation

12.42.1.1 [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

12.42.1.2 virtual [MetNoFimex::NetCDF_CDMWriter::~~NetCDF_CDMWriter](#) () [virtual]

12.42.2 Member Function Documentation

12.42.2.1 const **std::string**& [MetNoFimex::NetCDF_CDMWriter::getVariableName](#) (const **std::string** & *varName*) const

Returns:

the new name of a variable, eventually changed by the writers config

12.42.2.2 `const std::string& MetNoFimex::NetCDF_CDMWriter::getDimensionName (const std::string & dimName) const`

Returns:

the new name of a dimension, eventually changed by the writers config

12.42.2.3 `const CDMAAttribute& MetNoFimex::NetCDF_CDMWriter::getAttribute (const std::string & varName, const std::string & attName) const throw (CDMException)`

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

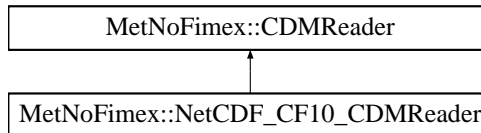
The documentation for this class was generated from the following file:

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

12.43 MetNoFimex::NetCDF_CF10_CDMReader Class Reference

```
#include <NetCDF_CF10_CDMReader.h>
```

Inheritance diagram for MetNoFimex::NetCDF_CF10_CDMReader::



Public Member Functions

- [NetCDF_CF10_CDMReader](#) (const **std::string** &fileName)
- virtual [~NetCDF_CF10_CDMReader](#) ()
- virtual const boost::shared_ptr< [Data](#) > [getDataSlice](#) (const **std::string** &varName, size_t unLimDimPos) throw (CDMException)

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

12.43.1 Constructor & Destructor Documentation

12.43.1.1 [MetNoFimex::NetCDF_CF10_CDMReader::NetCDF_CF10_CDMReader](#) (const **std::string** &fileName)

12.43.1.2 virtual [MetNoFimex::NetCDF_CF10_CDMReader::~~NetCDF_CF10_CDMReader](#) ()
[virtual]

12.43.2 Member Function Documentation

12.43.2.1 virtual const boost::shared_ptr<[Data](#)> [MetNoFimex::NetCDF_CF10_CDMReader::getDataSlice](#) (const **std::string** & varName, size_t unLimDimPos) throw (CDMException) [virtual]

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

This function 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](#).

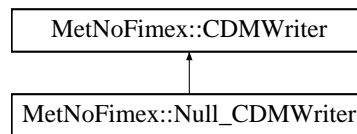
The documentation for this class was generated from the following file:

- [include/fimex/NetCDF_CF10_CDMReader.h](#)

12.44 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](#) ()

12.44.1 Detailed Description

[CDMWriter](#) does all operations as the [NetCDF_CDMWriter](#), except writing to the file. This class is useful for performance tests.

12.44.2 Constructor & Destructor Documentation

12.44.2.1 [MetNoFimex::Null_CDMWriter::Null_CDMWriter](#) (const boost::shared_ptr< [CDMReader](#) > *cdmReader*, const **std::string** & *outputFile*)

12.44.2.2 virtual [MetNoFimex::Null_CDMWriter::~~Null_CDMWriter](#) () [virtual]

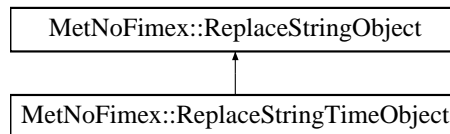
The documentation for this class was generated from the following file:

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

12.45 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

12.45.1 Detailed Description

Interface for objects which might be converted to different strings

12.45.2 Constructor & Destructor Documentation

12.45.2.1 virtual `MetNoFimex::ReplaceStringObject::~~ReplaceStringObject ()` [pure virtual]

12.45.3 Member Function Documentation

12.45.3.1 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](#).

12.45.3.2 virtual void `MetNoFimex::ReplaceStringObject::setFormatString (const std::string & format)` [pure virtual]

set the formatting string for this object

Implemented in [MetNoFimex::ReplaceStringTimeObject](#).

12.45.3.3 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](#).

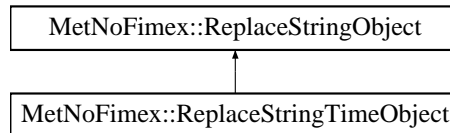
The documentation for this class was generated from the following file:

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

12.46 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<< \(std::ostream &s, const ReplaceStringTimeObject &rsto\)](#)

12.46.1 Constructor & Destructor Documentation

12.46.1.1 [MetNoFimex::ReplaceStringTimeObject::ReplaceStringTimeObject \(\)](#) [inline]

12.46.1.2 [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

12.46.1.3 [virtual MetNoFimex::ReplaceStringTimeObject::~~ReplaceStringTimeObject \(\)](#)
[inline, virtual]

12.46.2 Member Function Documentation

12.46.2.1 [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](#).

12.46.2.2 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](#).

12.46.2.3 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](#).

12.46.3 Friends And Related Function Documentation

12.46.3.1 std::ostream& operator<< (std::ostream & *s*, const ReplaceStringTimeObject & *rsto*) [friend]

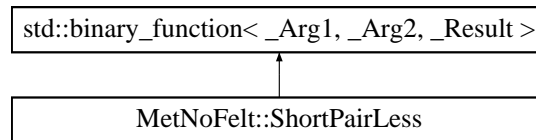
The documentation for this class was generated from the following file:

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

12.47 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

12.47.1 Detailed Description

comparison operator for `pair<short, short>` used for levelPairs

12.47.2 Member Function Documentation

12.47.2.1 `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`

12.48 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](#) ()

12.48.1 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 $x0 = i * (v1-v0) * \text{unit}$ 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?)

12.48.2 Constructor & Destructor Documentation

12.48.2.1 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

12.48.2.2 `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

12.48.2.3 `virtual MetNoFimex::SpatialAxisSpec::~~SpatialAxisSpec () [inline, virtual]`

12.48.3 Member Function Documentation

12.48.3.1 `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

12.48.3.2 `void MetNoFimex::SpatialAxisSpec::setStartEnd (double start, double end) [inline]`

12.48.3.3 `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](#)

12.49 KDTree::squared_difference< _Tp, _Dist > Struct Template Reference

```
#include <function.hpp>
```

Public Types

- typedef `_Dist` [distance_type](#)

Public Member Functions

- [distance_type operator\(\)](#) (const `_Tp` &__a, const `_Tp` &__b) const

```
template<typename _Tp, typename _Dist> struct KDTree::squared_difference< _Tp, _Dist >
```

12.49.1 Member Typedef Documentation

12.49.1.1 `template<typename _Tp, typename _Dist> typedef _Dist KDTree::squared_difference< _Tp, _Dist >::distance_type`

12.49.2 Member Function Documentation

12.49.2.1 `template<typename _Tp, typename _Dist> distance_type KDTree::squared_difference< _Tp, _Dist >::operator() (const _Tp & __a, const _Tp & __b) const` [`inline`]

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

- `include/kdtree++/function.hpp`

12.50 KDTree::squared_difference_counted< _Tp, _Dist > Struct Template Reference

```
#include <function.hpp>
```

Public Types

- typedef `_Dist` [distance_type](#)

Public Member Functions

- [squared_difference_counted\(\)](#)
- void [reset\(\)](#)
- long & [count\(\)](#) const
- [distance_type operator\(\)](#) (const `_Tp` &__a, const `_Tp` &__b) const

```
template<typename _Tp, typename _Dist> struct KDTree::squared_difference_counted< _Tp, _Dist >
```

12.50.1 Member Typedef Documentation

12.50.1.1 `template<typename _Tp, typename _Dist> typedef _Dist KDTree::squared_difference_counted< _Tp, _Dist >::distance_type`

12.50.2 Constructor & Destructor Documentation

12.50.2.1 `template<typename _Tp, typename _Dist> KDTree::squared_difference_counted< _Tp, _Dist >::squared_difference_counted() [inline]`

12.50.3 Member Function Documentation

12.50.3.1 `template<typename _Tp, typename _Dist> void KDTree::squared_difference_counted< _Tp, _Dist >::reset() [inline]`

12.50.3.2 `template<typename _Tp, typename _Dist> long& KDTree::squared_difference_counted< _Tp, _Dist >::count() const [inline]`

12.50.3.3 `template<typename _Tp, typename _Dist> distance_type KDTree::squared_difference_counted< _Tp, _Dist >::operator()(const _Tp & __a, const _Tp & __b) const [inline]`

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

- include/kdtree++/[function.hpp](#)

12.51 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)

12.51.1 Detailed Description

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

12.51.2 Constructor & Destructor Documentation

12.51.2.1 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

12.51.2.2 virtual MetNoFimex::TimeLevelDataSliceFetcher::~~TimeLevelDataSliceFetcher () [virtual]

12.51.3 Member Function Documentation

12.51.3.1 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](#)

12.52 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

12.52.1 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 $t0 = i * (v1-v0) * unit$ 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
```

12.52.2 Constructor & Destructor Documentation

12.52.2.1 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

12.52.2.2 `virtual MetNoFimex::TimeSpec::~~TimeSpec ()` [inline, virtual]

12.52.3 Member Function Documentation

12.52.3.1 `const std::vector<FimexTime>& MetNoFimex::TimeSpec::getTimeSteps () const`
[inline]

12.52.3.2 `const std::string& MetNoFimex::TimeSpec::getUnitString () const` [inline]

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

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

12.53 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\(\)\)](#)

12.53.1 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)

12.53.2 Constructor & Destructor Documentation

12.53.2.1 MetNoFimex::TimeUnit::TimeUnit () throw (CDMException)

initialize a timeUnit with a unit string

12.53.2.2 `MetNoFimex::TimeUnit::TimeUnit (const std::string & timeUnitString) throw (CDMException)`

12.53.2.3 `virtual MetNoFimex::TimeUnit::~~TimeUnit () [virtual]`

12.53.3 Member Function Documentation

12.53.3.1 `double MetNoFimex::TimeUnit::unitTime2epochSeconds (double unitTime) const`

calculate the epochSeconds for a time in the current unit

12.53.3.2 `double MetNoFimex::TimeUnit::epochSeconds2unitTime (double epochSeconds) const`

calculate the time in the current unit from the epoch

12.53.3.3 `FimexTime MetNoFimex::TimeUnit::unitTime2fimexTime (double unitTime) const throw (CDMException)`

calculate the time in a calendar form

12.53.3.4 `double MetNoFimex::TimeUnit::fimexTime2unitTime (const FimexTime & fiTime) const throw (CDMException)`

calculate the time in the current unit from the calendar form

Referenced by `fimexTime2unitTimeX()`.

12.53.3.5 `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()`.

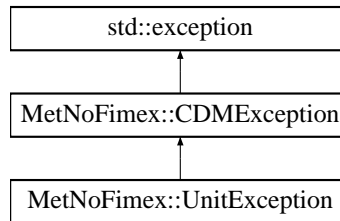
The documentation for this class was generated from the following file:

- `include/fimex/TimeUnit.h`

12.54 MetNoFimex::UnitException Class Reference

```
#include <Units.h>
```

Inheritance diagram for MetNoFimex::UnitException::



Public Member Functions

- [UnitException \(\)](#)
- [UnitException \(std::string message\)](#)

12.54.1 Constructor & Destructor Documentation

12.54.1.1 [MetNoFimex::UnitException::UnitException \(\)](#) [inline]

12.54.1.2 [MetNoFimex::UnitException::UnitException \(std::string message\)](#) [inline]

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

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

12.55 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)

12.55.1 Constructor & Destructor Documentation

12.55.1.1 MetNoFimex::Units::Units ()

initialization of unit handling, i.e. parsing of unit file etc if required

12.55.1.2 MetNoFimex::Units::Units (const Units & rhs)

12.55.1.3 virtual MetNoFimex::Units::~~Units () [virtual]

12.55.2 Member Function Documentation

12.55.2.1 Units& MetNoFimex::Units::operator= (const Units & rhs)

12.55.2.2 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

12.55.2.3 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

12.55.2.4 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](#)

12.56 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) const throw (CDMException)
- void [registerNamespace](#) (const **std::string** &prefix, const **std::string** &uri) throw (CDMException)

register a namespace for later xpath

12.56.1 Detailed Description

a tiny wrapper around libxml dom and xpath reader with xml::include

12.56.2 Constructor & Destructor Documentation

12.56.2.1 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

12.56.2.2 virtual MetNoFimex::XMLDoc::~~XMLDoc () [virtual]

12.56.3 Member Function Documentation

12.56.3.1 XPathObjPtr MetNoFimex::XMLDoc::getXPathObject (const **std::string** & *xpath*) 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

12.56.3.2 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 13

File Documentation

13.1 doxydoc.txt File Reference

13.2 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::CachedInterpolation](#)

13.3 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](#)

13.4 include/fimex/CDM.h File Reference

```
#include <map>
#include <vector>
#include <string>
#include <ostream>
#include <boost/regex.hpp>
#include "fimex/CDMAttribute.h"
#include "fimex/CDMVariable.h"
#include "fimex/CDMDimension.h"
#include "fimex/CDMException.h"
#include "fimex/CDMconstants.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::CDM](#)
Data structure of the Common Data Model.

13.5 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"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::CDMAttribute](#)

Functions

- **std::vector**< CDMAttribute > [MetNoFimex::projStringToAttributes](#) (**std::string** projStr)
convert a proj4 string to a list of CDMAttributes usable for CF-1.0 projection variable
- **std::string** [MetNoFimex::attributesToProjString](#) (const **std::vector**< CDMAttribute > &attrs)
convert attributes of a projection-variable to a projString

13.6 include/fimex/CDMconstants.h File Reference

Defines

- #define [MIFI_EARTH_RADIUS_M](#) 6371000

13.6.1 Define Documentation

13.6.1.1 #define MIFI_EARTH_RADIUS_M 6371000

13.7 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)

13.8 include/fimex/CDMDimension.h File Reference

```
#include <string>
#include <ostream>
#include "fimex/CDMNamedEntity.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::CDMDimension](#)

13.9 include/fimex/CDMException.h File Reference

```
#include <exception>
#include <string>
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::CDMException](#)

13.10 include/fimex/CDMExtractor.h File Reference

```
#include <boost/array.hpp>
#include <boost/shared_ptr.hpp>
#include "fimex/CDMReader.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::CDMExtractor](#)

13.11 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](#)

13.12 `include/fimex/CDMNamedEntity.h` File Reference

```
#include <string>
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::CDMNamedEntity](#)
- struct [MetNoFimex::CDMNameCompare](#)
- class [MetNoFimex::CDMNameEqual](#)

13.13 include/fimex/CDMReader.h File Reference

```
#include <boost/shared_ptr.hpp>
#include "fimex/CDM.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::CDMReader](#)
Basic interface for [CDM](#) reading and manipulation classes.

13.14 include/fimex/CDMTimeInterpolator.h File Reference

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

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::CDMTimeInterpolator](#)

13.15 include/fimex/CDMVariable.h File Reference

```
#include <string>
#include <vector>
#include <ostream>
#include "fimex/CDMAttribute.h"
#include "fimex/CDMDimension.h"
#include "fimex/CDMDataType.h"
#include "fimex/CDMNamedEntity.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::CDMVariable](#)

13.16 include/fimex/CDMWriter.h File Reference

```
#include <string>
#include <boost/shared_ptr.hpp>
#include "fimex/CDMReader.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::CDMWriter](#)

13.17 include/fimex/config.h File Reference

Defines

- #define [HAVE_BOOST](#)
- #define [HAVE_BOOST_PROGRAM_OPTIONS](#)
- #define [HAVE_BOOST_REGEX](#)
- #define [HAVE_BOOST_UNIT_TEST_FRAMEWORK](#)
- #define [HAVE_CEIL](#) 1
- #define [HAVE_DLFCN_H](#) 1
- #define [HAVE_FMOD](#) 1
- #define [HAVE_GRIBAPI_H](#) 1
- #define [HAVE_INTTYPES_H](#) 1
- #define [HAVE_LIBM](#) 1
- #define [HAVE_LIBMIC](#) 1
- #define [HAVE_LOG10](#) 1
- #define [HAVE_MEMORY_H](#) 1
- #define [HAVE_MEMSET](#) 1
- #define [HAVE_MILIB_MILIB_H](#) 1
- #define [HAVE_NAMESPACES](#)
- #define [HAVE_NCFILE_FILEFORMAT](#) 1
- #define [HAVE_NETCDF](#) 1
- #define [HAVE_OPENMP](#) 1
- #define [HAVE_POW](#) 1
- #define [HAVE_PROJ4](#) 1
- #define [HAVE_SQRT](#) 1
- #define [HAVE_STD](#)
- #define [HAVE_STDBOOL_H](#) 1
- #define [HAVE_STDINT_H](#) 1
- #define [HAVE_STDLIB_H](#) 1
- #define [HAVE_STL](#)
- #define [HAVE_STRINGS_H](#) 1
- #define [HAVE_STRING_H](#) 1
- #define [HAVE_STRSTR](#) 1
- #define [HAVE_SYS_STAT_H](#) 1
- #define [HAVE_SYS_TYPES_H](#) 1
- #define [HAVE_UDUNITS](#) 1
- #define [HAVE_UNISTD_H](#) 1
- #define [LSTAT_FOLLOWS_SLASHED_SYMLINK](#) 1
- #define [NETCDF_CPP_INCLUDE](#) "/usr/include/netcdfcpp.h"
- #define [NETCDF_C_INCLUDE](#) "/usr/include/netcdf.h"
- #define [PACKAGE](#) "fimex"
- #define [PACKAGE_BUGREPORT](#) "heiko.klein@met.no"
- #define [PACKAGE_NAME](#) "fimex"
- #define [PACKAGE_STRING](#) "fimex 0.14"
- #define [PACKAGE_TARNAME](#) "fimex"
- #define [PACKAGE_VERSION](#) "0.14"
- #define [STDC_HEADERS](#) 1
- #define [VERSION](#) "0.14"

13.17.1 Define Documentation

- 13.17.1.1 `#define HAVE_BOOST`
- 13.17.1.2 `#define HAVE_BOOST_PROGRAM_OPTIONS`
- 13.17.1.3 `#define HAVE_BOOST_REGEX`
- 13.17.1.4 `#define HAVE_BOOST_UNIT_TEST_FRAMEWORK`
- 13.17.1.5 `#define HAVE_CEIL 1`
- 13.17.1.6 `#define HAVE_DLFCN_H 1`
- 13.17.1.7 `#define HAVE_FMOD 1`
- 13.17.1.8 `#define HAVE_GRIBAPI_H 1`
- 13.17.1.9 `#define HAVE_INTTYPES_H 1`
- 13.17.1.10 `#define HAVE_LIBM 1`
- 13.17.1.11 `#define HAVE_LIBMIC 1`
- 13.17.1.12 `#define HAVE_LOG10 1`
- 13.17.1.13 `#define HAVE_MEMORY_H 1`
- 13.17.1.14 `#define HAVE_MEMSET 1`
- 13.17.1.15 `#define HAVE_MILIB_MILIB_H 1`
- 13.17.1.16 `#define HAVE_NAMESPACES`
- 13.17.1.17 `#define HAVE_NCFILE_FILEFORMAT 1`
- 13.17.1.18 `#define HAVE_NETCDF 1`
- 13.17.1.19 `#define HAVE_OPENMP 1`
- 13.17.1.20 `#define HAVE_POW 1`
- 13.17.1.21 `#define HAVE_PROJ4 1`
- 13.17.1.22 `#define HAVE_SQRT 1`
- 13.17.1.23 `#define HAVE_STD`
- 13.17.1.24 `#define HAVE_STDBOOL_H 1`
- 13.17.1.25 `#define HAVE_STDINT_H 1`
- 13.17.1.26 `#define HAVE_STDLIB_H 1`
- 13.17.1.27 `#define HAVE_STL`
- 13.17.1.28 `#define HAVE_STRING_H 1`
- 13.17.1.29 `#define HAVE_STRINGS_H 1`
- 13.17.1.30 `#define HAVE_STRSTR 1`

13.18 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) 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 dataslice from another [Data](#) object

13.19 include/fimex/DataImpl.h File Reference

```
#include <typeinfo>
#include <boost/shared_ptr.hpp>
#include <string>
#include <sstream>
#include <iostream>
#include <cmath>
#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, size_t length, 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 T1, typename T2>`
`boost::shared_array< T1 > MetNoFimex::convertArrayType (const boost::shared_array< T2 > &inData, size_t length, double oldFill, double oldScale, double oldOffset, double newFill, double newScale, double newOffset)`

13.20 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](#)

13.21 include/fimex/Felt_Array.h File Reference

```
#include <string>
#include <set>
#include <vector>
#include <map>
#include <ctime>
#include <boost/array.hpp>
#include "fimex/Felt_File_Error.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

- time_t [MetNoFelt::index16toTime](#) (const boost::array< short, 16 > &idx)
- pair< short, short > [MetNoFelt::index16toLevelPair](#) (const boost::array< short, 16 > &idx)

13.22 include/fimex/Felt_File.h File Reference

```
#include <ctime>
#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"
```

Namespaces

- namespace [MetNoFelt](#)

Classes

- class [MetNoFelt::Felt_File](#)
Felt File access.

13.23 include/fimex/Felt_File_Error.h File Reference

```
#include <exception>
#include <string>
```

Namespaces

- namespace [MetNoFelt](#)

Classes

- class [MetNoFelt::Felt_File_Error](#)

13.24 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"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::FeltCDMReader](#)

13.25 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](#) ()

13.26 include/fimex/GribApiCDMWriter.h File Reference

```
#include "fimex/CDMWriter.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::GribApiCDMWriter](#)

13.27 include/fimex/GribApiCDMWriter_Impl1.h File Reference

```
#include "fimex/GribApiCDMWriter_ImplAbstract.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::GribApiCDMWriter_Impl1](#)

13.28 include/fimex/GribApiCDMWriter_Impl2.h File Reference

```
#include "fimex/GribApiCDMWriter_ImplAbstract.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::GribApiCDMWriter_Impl2](#)

13.29 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](#)

13.30 include/fimex/interpolation.h File Reference

```
#include <proj_api.h>
#include <math.h>
```

Defines

- #define [PI](#) 3.1415926535897932384626433832795
- #define [MIFI_NEAREST_NEIGHBOR](#) 0
interpolation method
- #define [MIFI_BILINEAR](#) 1
interpolation method
- #define [MIFI_BICUBIC](#) 2
interpolation method
- #define [MIFI_COORD_NN](#) 3
interpolation method
- #define [MIFI_COORD_NN_KD](#) 4
interpolation method
- #define [MIFI_VECTOR_KEEP_SIZE](#) 0
vector projection flag
- #define [MIFI_VECTOR_RESIZE](#) 1
vector projection flag
- #define [MIFI_UNDEFINED_F](#) (nanf(""))
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)
- 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 $(x,y) \Rightarrow (x_{proj}), (y_{proj})$ 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 $(x,y) \Rightarrow (x_{proj}), (y_{proj})$ 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)

13.30.1 Define Documentation

13.30.1.1 #define MIFI_BICUBIC 2

interpolation method

flag for bicubic interpolation

13.30.1.2 #define MIFI_BILINEAR 1

interpolation method

flag for bilinear interpolation

13.30.1.3 #define MIFI_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

13.30.1.4 #define MIFI_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

13.30.1.5 #define MIFI_DEBUG 0

debug flag

13.30.1.6 #define MIFI_ERROR -1

return code, error

13.30.1.7 #define MIFI_LATITUDE 2

latitude projection axis in degrees

13.30.1.8 #define MIFI_LONGITUDE 1

longitude projection axis in degrees

13.30.1.9 #define MIFI_NEAREST_NEIGHBOR 0

interpolation method

flag for nearest neighbor interpolation

13.30.1.10 #define MIFI_OK 1

return code, ok

13.30.1.11 #define MIFI_PROJ_AXIS 0

projection axis in m-equivalent

13.30.1.12 #define MIFI_UNDEFINED_D (nan(""))

undefined value for doubles

13.30.1.13 #define MIFI_UNDEFINED_F (nanf(""))

undefined value for floats

13.30.1.14 #define MIFI_VECTOR_KEEP_SIZE 0

vector projection flag

new size will be like old size

13.30.1.15 #define MIFI_VECTOR_RESIZE 1

vector projection flag

vector might change size with projection

13.30.1.16 #define PI 3.1415926535897932384626433832795**13.30.2 Function Documentation****13.30.2.1 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

13.30.2.2 `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

13.30.2.3 `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/javax/media/>

13.30.2.4 `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:

$$\begin{array}{cc} s00 & s01 \\ & \cdot < yfrac \end{array}$$

```

s10    s11
   ^
   xfrac

```

See also:

<http://java.sun.com/products/java-media/jai/forDevelopers/jai-apidocs/javax/media/>

Warning:

if any of the 4 used values of *infield* is undefined or outside of *infield*, the return value will be undefined

13.30.2.5 `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

13.30.2.6 `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 $o(x) = in(a) + x * (in(a) - in(b)) / (a - b)$

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*

13.30.2.7 `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 reproject 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

13.30.2.8 `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](#)

13.30.2.9 `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_NEAREST_NEIGHBOR MIFI_BILINEAR MIFI_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

13.30.2.10 `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

13.30.2.11 `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

13.30.2.12 `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

13.30.2.13 `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

13.30.2.14 `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

13.30.2.15 `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

13.31 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)

13.31.1 Define Documentation

13.31.1.1 #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

13.32 include/fimex/NcmlCDMReader.h File Reference

```
#include "CDMReader.h"  
#include "boost/noncopyable.hpp"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::NcmlCDMReader](#)

13.33 include/fimex/NetCDF_CDMWriter.h File Reference

```
#include "fimex/CDMWriter.h"  
#include "fimex/CDM.h"  
#include "fimex/config.h"  
#include <map>  
#include <string>
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::NetCDF_CDMWriter](#)

13.34 include/fimex/NetCDF_CF10_CDMReader.h File Reference

```
#include "fimex/config.h"  
#include "fimex/CDMReader.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::NetCDF_CF10_CDMReader](#)

13.35 include/fimex/NetCDF_Utils.h File Reference

```
#include <boost/shared_ptr.hpp>
#include "fimex/config.h"
#include "fimex/CDMDataType.h"
#include "fimex/Data.h"
```

Namespaces

- namespace [MetNoFimex](#)

Functions

- NcType [MetNoFimex::cdmDataType2ncType](#) (CDMDataType dt)
- CDMDataType [MetNoFimex::ncType2cdmDataType](#) (NcType dt)
- boost::shared_ptr< Data > [MetNoFimex::ncValues2Data](#) (NcValues *values, NcType dt, size_t length)

13.36 include/fimex/Null_CDMWriter.h File Reference

```
#include "fimex/CDMWriter.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::Null_CDMWriter](#)

13.37 include/fimex/ReplaceStringObject.h File Reference

```
#include <iostream>
#include <string>
#include <vector>
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::ReplaceStringObject](#)

13.38 include/fimex/ReplaceStringTimeObject.h File Reference

```
#include "fimex/ReplaceStringObject.h"  
#include <ctime>
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::ReplaceStringTimeObject](#)

13.39 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](#)

13.40 include/fimex/TimeLevelDataSliceFetcher.h File Reference

```
#include <boost/shared_ptr.hpp>
#include "fimex/CDMReader.h"
#include "fimex/Data.h"
#include "fimex/Logger.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::TimeLevelDataSliceFetcher](#)
read a slice of a given time/level combination from a cdmReader

13.41 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](#)

13.42 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)`

13.43 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)

13.44 include/fimex/Utils.h File Reference

```
#include <vector>
#include <sstream>
#include "fimex/CDMException.h"
```

Namespaces

- namespace [MetNoFimex](#)

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=",")
- **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<typename T>
T [MetNoFimex::string2type](#) (**std::string** s)
- template<typename T>
std::vector< T > [MetNoFimex::tokenizeDotted](#) (const **std::string** &str, const **std::string** &delimiter=",") throw (CDMException)

13.45 include/fimex/XMLDoc.h File Reference

```
#include <boost/utility.hpp>
#include <boost/shared_ptr.hpp>
#include <string>
#include <libxml/tree.h>
#include <libxml/xpath.h>
#include "fimex/CDMException.h"
#include "fimex/XMLDoc.h"
```

Namespaces

- namespace [MetNoFimex](#)

Classes

- class [MetNoFimex::XMLDoc](#)

Typedefs

- 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)

13.46 include/kdtree++/allocator.hpp File Reference

```
#include <cstddef>
#include "node.hpp"
```

Namespaces

- namespace [KDTree](#)

Classes

- class [KDTree::_Alloc_base<_Tp, _Alloc >](#)
- class [KDTree::_Alloc_base<_Tp, _Alloc >::NoLeakAlloc](#)

13.46.1 Detailed Description

Defines the allocator interface as used by the [KDTree](#) class.

Author:

Martin F. Krafft <libkdtree@pobox.madduck.net>

13.47 include/kdtree++/function.hpp File Reference

```
#include <cstdint>
```

Namespaces

- namespace [KDTree](#)

Classes

- struct [KDTree::_Bracket_accessor<_Val>](#)
- struct [KDTree::always_true<_Tp>](#)
- struct [KDTree::squared_difference<_Tp, _Dist>](#)
- struct [KDTree::squared_difference_counted<_Tp, _Dist>](#)

13.47.1 Detailed Description

Defines the various functors and interfaces used for [KDTree](#).

Author:

Martin F. Krafft <libkdtree@pobox.madduck.net>
Sylvain Bougerel <sylvain.bougerel.devel@gmail.com>

13.48 include/kdtree++/iterator.hpp File Reference

```
#include <iterator>
#include <kdtree++/node.hpp>
```

Namespaces

- namespace [KDTree](#)

Classes

- class [KDTree::_Base_iterator](#)
- class [KDTree::_Iterator<_Val, _Ref, _Ptr>](#)

Functions

- `template<typename _Val, typename _Ref, typename _Ptr>`
`bool KDTree::operator==(_Iterator<_Val, _Ref, _Ptr> const &, _Iterator<_Val, _Ref, _Ptr> const &)`
- `template<typename _Val>`
`bool KDTree::operator==(_Iterator<_Val, const _Val &, const _Val * > const &, _Iterator<_Val, _Val &, _Val * > const &)`
- `template<typename _Val>`
`bool KDTree::operator==(_Iterator<_Val, _Val &, _Val * > const &, _Iterator<_Val, const _Val &, const _Val * > const &)`
- `template<typename _Val, typename _Ref, typename _Ptr>`
`bool KDTree::operator!=(_Iterator<_Val, _Ref, _Ptr> const &, _Iterator<_Val, _Ref, _Ptr> const &)`
- `template<typename _Val>`
`bool KDTree::operator!=(_Iterator<_Val, const _Val &, const _Val * > const &, _Iterator<_Val, _Val &, _Val * > const &)`
- `template<typename _Val>`
`bool KDTree::operator!=(_Iterator<_Val, _Val &, _Val * > const &, _Iterator<_Val, const _Val &, const _Val * > const &)`

13.48.1 Detailed Description

Defines interfaces for iterators as used by the [KDTree](#) class.

Author:

Martin F. Krafft <libkdtree@pobox.madduck.net>

13.49 include/kdtree++/kdtree.hpp File Reference

```
#include <vector>
#include <algorithm>
#include <cmath>
#include <cstdlib>
#include <cassert>
#include "function.hpp"
#include "allocator.hpp"
#include "iterator.hpp"
#include "node.hpp"
#include "region.hpp"
```

Namespaces

- namespace [KDTree](#)

Classes

- class [KDTree::KDTree< __K, _Val, _Acc, _Dist, _Cmp, _Alloc >](#)

Defines

- #define [KDTREE_VERSION](#) 700
- #define [KDTREE_LIB_VERSION](#) "0_7_0"

13.49.1 Detailed Description

Defines the interface for the [KDTree](#) class.

Author:

Martin F. Krafft <libkdtree@pobox.madduck.net>

Paul Harris figured this stuff out (below) Notes: This is similar to a binary tree, but its not the same. There are a few important differences:

- * Each level is sorted by a different criteria (this is fundamental to the design).
- * It is possible to have children IDENTICAL to its parent in BOTH branches This is different to a binary tree, where identical children are always to the right So, [KDTree](#) has the relationships: * The left branch is <= its parent (in binary tree, this relationship is a plain <) * The right branch is <= its parent (same as binary tree)

This is done for mostly for performance. Its a LOT easier to maintain a consistent tree if we use the <= relationship. Note that this relationship only makes a difference when searching for an exact item with find() or find_exact, other search, erase and insert functions don't notice the difference.

In the case of binary trees, you can safely assume that the next identical item will be the child leaf, but in the case of [KDTree](#), the next identical item might be a long way down a subtree, because of the various different sort criteria.

So erase()ing a node from a [KDTree](#) could require serious and complicated tree rebalancing to maintain consistency... IF we required binary-tree-like relationships.

This has no effect on insert(s), a < test is good enough to keep consistency.

It has an effect on find() searches: * Instead of using compare(child,node) for a < relationship and following 1 branch, we must use !compare(node,child) for a <= relationship, and test BOTH branches, as we could potentially go down both branches.

It has no real effect on bounds-based searches (like find_nearest, find_within_range) as it compares vs a boundary and would follow both branches if required.

This has no real effect on erase(s), a < test is good enough to keep consistency.

13.49.2 Define Documentation

13.49.2.1 `#define KDTREE_LIB_VERSION "0_7_0"`

13.49.2.2 `#define KDTREE_VERSION 700`

13.50 include/kdtree++/node.hpp File Reference

```
#include <cstdint>
#include <cmath>
```

Namespaces

- namespace [KDTree](#)

Classes

- struct [KDTree::_Node_base](#)
- struct [KDTree::_Node<_Val>](#)
- class [KDTree::_Node_compare<_Val, _Acc, _Cmp>](#)

Functions

- template<typename _ValA, typename _ValB, typename _Cmp, typename _Acc>
bool [KDTree::_S_node_compare](#) (const size_t __dim, const _Cmp &__cmp, const _Acc &__acc, const _ValA &__a, const _ValB &__b)
- template<typename _ValA, typename _ValB, typename _Dist, typename _Acc>
_Dist::distance_type [KDTree::_S_node_distance](#) (const size_t __dim, const _Dist &__dist, const _Acc &__acc, const _ValA &__a, const _ValB &__b)
- template<typename _ValA, typename _ValB, typename _Dist, typename _Acc>
_Dist::distance_type [KDTree::_S_accumulate_node_distance](#) (const size_t __dim, const _Dist &__dist, const _Acc &__acc, const _ValA &__a, const _ValB &__b)
- template<typename _Val, typename _Cmp, typename _Acc>
_Node_base * [KDTree::_S_node_descend](#) (const size_t __dim, const _Cmp &__cmp, const _Acc &__acc, const _Val &__val, const _Node_base *__node)
- template<class SearchVal, typename _Val, typename _Cmp, typename _Acc, typename _Dist, typename _Predicate>
std::pair< const _Node< _Val > *, **std::pair**< size_t, typename _Dist::distance_type > >
[KDTree::_S_node_nearest](#) (const size_t __k, size_t __dim, SearchVal const &__val, const _Node< _Val > *__node, const _Node_base *__end, const _Node< _Val > *__best, typename _Dist::distance_type __max, const _Cmp &__cmp, const _Acc &__acc, const _Dist &__dist, _Predicate __p)

13.50.1 Detailed Description

Defines interfaces for nodes as used by the [KDTree](#) class.

Author:

Martin F. Krafft <libkdtree@pobox.madduck.net>

13.51 include/kdtree++/region.hpp File Reference

```
#include <cstdint>
#include <kdtree++/node.hpp>
```

Namespaces

- namespace [KDTree](#)

Classes

- struct [KDTree::_Region](#)< __K, _Val, _SubVal, _Acc, _Cmp >

13.51.1 Detailed Description

Defines the interface of the `_Region` class.

Author:

Martin F. Krafft <libkdtree@pobox.madduck.net>

Index

- ~CDM
 - MetNoFimex::CDM, 82
- ~CDMAttribute
 - MetNoFimex::CDMAttribute, 93
- ~CDMDimension
 - MetNoFimex::CDMDimension, 95
- ~CDMException
 - MetNoFimex::CDMException, 97
- ~CDMExtractor
 - MetNoFimex::CDMExtractor, 98
- ~CDMInterpolator
 - MetNoFimex::CDMInterpolator, 101
- ~CDMNameEqual
 - MetNoFimex::CDMNameEqual, 106
- ~CDMNamedEntity
 - MetNoFimex::CDMNamedEntity, 105
- ~CDMReader
 - MetNoFimex::CDMReader, 108
- ~CDMTimeInterpolator
 - MetNoFimex::CDMTimeInterpolator, 110
- ~CDMVariable
 - MetNoFimex::CDMVariable, 113
- ~CDMWriter
 - MetNoFimex::CDMWriter, 115
- ~CachedInterpolation
 - MetNoFimex::CachedInterpolation, 76
- ~CachedVectorReprojection
 - MetNoFimex::CachedVectorReprojection, 78
- ~Data
 - MetNoFimex::Data, 117
- ~DataImpl
 - MetNoFimex::DataImpl, 123
- ~DataTypeChanger
 - MetNoFimex::DataTypeChanger, 129
- ~FeltCDMReader
 - MetNoFimex::FeltCDMReader, 141
- ~FeltParameters
 - MetNoFelt::FeltParameters, 142
- ~Felt_Array
 - MetNoFelt::Felt_Array, 132
- ~Felt_File
 - MetNoFelt::Felt_File, 137
- ~Felt_File_Error
 - MetNoFelt::Felt_File_Error, 140
- ~GribApiCDMWriter
 - MetNoFimex::GribApiCDMWriter, 147
- ~GribApiCDMWriter_Impl1
 - MetNoFimex::GribApiCDMWriter_Impl1, 148
- ~GribApiCDMWriter_Impl2
 - MetNoFimex::GribApiCDMWriter_Impl2, 150
- ~GribApiCDMWriter_ImplAbstract
 - MetNoFimex::GribApiCDMWriter_ImplAbstract, 153
- ~KDTree
 - KDTree::KDTree, 163
- ~Logger
 - MetNoFimex::Logger, 187
- ~NcmlCDMReader
 - MetNoFimex::NcmlCDMReader, 190
- ~NetCDF_CDMWriter
 - MetNoFimex::NetCDF_CDMWriter, 191
- ~NetCDF_CF10_CDMReader
 - MetNoFimex::NetCDF_CF10_CDMReader, 193
- ~NoLeakAlloc
 - KDTree::_Alloc_base::NoLeakAlloc, 58
- ~Null_CDMWriter
 - MetNoFimex::Null_CDMWriter, 195
- ~ReplaceStringObject
 - MetNoFimex::ReplaceStringObject, 196
- ~ReplaceStringTimeObject
 - MetNoFimex::ReplaceStringTimeObject, 198
- ~SpatialAxisSpec
 - MetNoFimex::SpatialAxisSpec, 202
- ~TimeLevelDataSliceFetcher
 - MetNoFimex::TimeLevelDataSliceFetcher, 205
- ~TimeSpec
 - MetNoFimex::TimeSpec, 207
- ~TimeUnit
 - MetNoFimex::TimeUnit, 209
- ~Units
 - MetNoFimex::Units, 211
- ~XMLDoc
 - MetNoFimex::XMLDoc, 213
- _Alloc_base
 - KDTree::_Alloc_base, 56
- _Base

- KDTree::KDTree, 160
- _Base_const_ptr
 - KDTree::_Base_iterator, 59
 - KDTree::_Node_base, 68
 - KDTree::KDTree, 160
- _Base_iterator
 - KDTree::_Base_iterator, 59
- _Base_ptr
 - KDTree::_Alloc_base, 56
 - KDTree::_Node_base, 68
 - KDTree::KDTree, 160
- _CenterPt
 - KDTree::_Region, 72
- _Iterator
 - KDTree::_Iterator, 63
- _Link_const_type
 - KDTree::_Iterator, 63
 - KDTree::KDTree, 162
- _Link_type
 - KDTree::_Node, 66
 - KDTree::KDTree, 162
- _M_acc
 - KDTree::_Region, 74
 - KDTree::KDTree, 185
- _M_allocate_node
 - KDTree::_Alloc_base, 56
- _M_check_children
 - KDTree::KDTree, 172
- _M_check_node
 - KDTree::KDTree, 172
- _M_cmp
 - KDTree::_Region, 74
 - KDTree::KDTree, 185
- _M_construct_node
 - KDTree::_Alloc_base, 56
- _M_count
 - KDTree::KDTree, 185
- _M_count_within_range
 - KDTree::KDTree, 177
- _M_deallocate_node
 - KDTree::_Alloc_base, 56
- _M_decrement
 - KDTree::_Base_iterator, 60
- _M_delete_node
 - KDTree::KDTree, 184
- _M_destroy_node
 - KDTree::_Alloc_base, 57
- _M_dist
 - KDTree::KDTree, 186
- _M_empty_initialise
 - KDTree::KDTree, 173
- _M_erase
 - KDTree::KDTree, 174
- _M_erase_subtree
 - KDTree::KDTree, 175
- _M_find
 - KDTree::KDTree, 175
- _M_find_exact
 - KDTree::KDTree, 176
- _M_find_within_range
 - KDTree::KDTree, 177
- _M_get_erase_replacement
 - KDTree::KDTree, 174
- _M_get_j_max
 - KDTree::KDTree, 175
- _M_get_j_min
 - KDTree::KDTree, 175
- _M_get_leftmost
 - KDTree::KDTree, 179
- _M_get_rightmost
 - KDTree::KDTree, 179
- _M_get_root
 - KDTree::KDTree, 178
- _M_header
 - KDTree::KDTree, 184
- _M_high_bounds
 - KDTree::_Region, 74
- _M_increment
 - KDTree::_Base_iterator, 59
- _M_insert
 - KDTree::KDTree, 174
- _M_insert_left
 - KDTree::KDTree, 173
- _M_insert_right
 - KDTree::KDTree, 173
- _M_left
 - KDTree::_Node_base, 68
- _M_low_bounds
 - KDTree::_Region, 73
- _M_matches_node
 - KDTree::KDTree, 177
- _M_matches_node_in_d
 - KDTree::KDTree, 176
- _M_matches_node_in_other_ds
 - KDTree::KDTree, 176
- _M_new_node
 - KDTree::KDTree, 184
- _M_node
 - KDTree::_Base_iterator, 60
- _M_node_allocator
 - KDTree::_Alloc_base, 57
- _M_optimise
 - KDTree::KDTree, 178
- _M_parent
 - KDTree::_Node_base, 68
- _M_right
 - KDTree::_Node_base, 68
- _M_root

- KDTree::KDTree, 184
- _M_set_leftmost
 - KDTree::KDTree, 179
- _M_set_rightmost
 - KDTree::KDTree, 180
- _M_set_root
 - KDTree::KDTree, 179
- _M_value
 - KDTree::_Node, 66
- _M_visit_within_range
 - KDTree::KDTree, 177
- _Node
 - KDTree::_Node, 66
- _Node_
 - KDTree::_Alloc_base, 56
- _Node_base
 - KDTree::_Node_base, 68
- _Node_compare
 - KDTree::_Node_compare, 70
- _Node_compare_
 - KDTree::KDTree, 162
- _Region
 - KDTree::_Region, 72
- _Region_
 - KDTree::KDTree, 162
- _S_accumulate_node_distance
 - KDTree, 42
- _S_is_leaf
 - KDTree::KDTree, 183
- _S_left
 - KDTree::KDTree, 181, 182
- _S_maximum
 - KDTree::_Node_base, 68
 - KDTree::KDTree, 183
- _S_minimum
 - KDTree::_Node_base, 68
 - KDTree::KDTree, 183
- _S_node_compare
 - KDTree, 42
- _S_node_descend
 - KDTree, 42
- _S_node_distance
 - KDTree, 42
- _S_node_nearest
 - KDTree, 43
- _S_parent
 - KDTree::KDTree, 180
- _S_right
 - KDTree::KDTree, 182
- _S_set_left
 - KDTree::KDTree, 181
- _S_set_parent
 - KDTree::KDTree, 181
- _S_set_right
 - KDTree::KDTree, 182
- _S_value
 - KDTree::KDTree, 183
- _Self
 - KDTree::_Iterator, 63
- addAttribute
 - MetNoFimex::CDM, 86
- addDimension
 - MetNoFimex::CDM, 84
- addIdent19
 - MetNoFelt::Felt_Array, 134
- addInformationByIndex
 - MetNoFelt::Felt_Array, 132
- addOrReplaceAttribute
 - MetNoFimex::CDM, 86
- addVariable
 - MetNoFimex::CDM, 82
- allocator_type
 - KDTree::_Alloc_base, 56
 - KDTree::KDTree, 160
- ANY_ARRAY
 - MetNoFelt, 46
- ANY_ARRAY20
 - MetNoFelt, 46
- ANY_VALUE
 - MetNoFelt, 46
- areConvertible
 - MetNoFimex::Units, 211
- as
 - MetNoFimex::DataImpl, 123, 124
- asBase
 - MetNoFimex::DataImpl, 123
- asChar
 - MetNoFimex::Data, 118
 - MetNoFimex::DataImpl, 124
- asConstChar
 - MetNoFimex::Data, 118
 - MetNoFimex::DataImpl, 124
- asConstDouble
 - MetNoFimex::Data, 119
 - MetNoFimex::DataImpl, 125
- asConstFloat
 - MetNoFimex::Data, 118
 - MetNoFimex::DataImpl, 124
- asConstInt
 - MetNoFimex::Data, 118
 - MetNoFimex::DataImpl, 124
- asConstShort
 - MetNoFimex::Data, 118
 - MetNoFimex::DataImpl, 124
- asDouble
 - MetNoFimex::Data, 119
 - MetNoFimex::DataImpl, 125

- asFloat
 - MetNoFimex::Data, 119
 - MetNoFimex::DataImpl, 125
- asInt
 - MetNoFimex::Data, 118
 - MetNoFimex::DataImpl, 124
- asShort
 - MetNoFimex::Data, 118
 - MetNoFimex::DataImpl, 124
- asString
 - MetNoFimex::Data, 119
 - MetNoFimex::DataImpl, 125
- attributesToProjString
 - MetNoFimex, 50
- AttrVec
 - MetNoFimex::CDM, 82
- begin
 - KDTree::KDTree, 166
- bytes_for_one
 - MetNoFimex::Data, 117
 - MetNoFimex::DataImpl, 123
- CachedInterpolation
 - MetNoFimex::CachedInterpolation, 76
- CachedVectorReprojection
 - MetNoFimex::CachedVectorReprojection, 78
- CDM
 - MetNoFimex::CDM, 82
- cdm
 - MetNoFimex::CDMReader, 109
- CDM_CHAR
 - MetNoFimex, 49
- CDM_DOUBLE
 - MetNoFimex, 49
- CDM_FLOAT
 - MetNoFimex, 49
- CDM_INT
 - MetNoFimex, 49
- CDM_NAT
 - MetNoFimex, 49
- CDM_SHORT
 - MetNoFimex, 49
- CDM_STRING
 - MetNoFimex, 49
- CDMAttribute
 - MetNoFimex::CDMAttribute, 92, 93
- CDMconstants.h
 - MIFI_EARTH_RADIUS_M, 220
- CDMDataType
 - MetNoFimex, 49
- cdmDataType2ncType
 - MetNoFimex, 50
- CDMDimension
 - MetNoFimex::CDMDimension, 95
- CDMException
 - MetNoFimex::CDMException, 97
- CDMExtractor
 - MetNoFimex::CDMExtractor, 98
- CDMInterpolator
 - MetNoFimex::CDMInterpolator, 101
- CDMNameEqual
 - MetNoFimex::CDMNameEqual, 106
- CDMReader
 - MetNoFimex::CDMReader, 108
- cdmReader
 - MetNoFimex::CDMWriter, 115
 - MetNoFimex::GribApiCDMWriter_ - ImplAbstract, 155
- CDMTimeInterpolator
 - MetNoFimex::CDMTimeInterpolator, 110
- CDMVariable
 - MetNoFimex::CDMVariable, 113
- CDMWriter
 - MetNoFimex::CDMWriter, 115
- changeDataType
 - MetNoFimex::CDMExtractor, 100
- changeProjection
 - MetNoFimex::CDMInterpolator, 101, 102
- changeTimeAxis
 - MetNoFimex::CDMTimeInterpolator, 110
- check_tree
 - KDTree::KDTree, 172
- checkDimension
 - MetNoFimex::CDMVariable, 113
- checkVariableAttribute
 - MetNoFimex::CDM, 84
- clear
 - KDTree::KDTree, 164
- config.h
 - HAVE_BOOST, 233
 - HAVE_BOOST_PROGRAM_OPTIONS, 233
 - HAVE_BOOST_REGEX, 233
 - HAVE_BOOST_UNIT_TEST_ - FRAMEWORK, 233
 - HAVE_CEIL, 233
 - HAVE_DLFCN_H, 233
 - HAVE_FMOD, 233
 - HAVE_GRIBAPI_H, 233
 - HAVE_INTTYPES_H, 233
 - HAVE_LIBM, 233
 - HAVE_LIBMIC, 233
 - HAVE_LOG10, 233
 - HAVE_MEMORY_H, 233
 - HAVE_MEMSET, 233
 - HAVE_MILIB_MILIB_H, 233
 - HAVE_NAMESPACES, 233
 - HAVE_NCFILE_FILEFORMAT, 233

- HAVE_NETCDF, 233
- HAVE_OPENMP, 233
- HAVE_POW, 233
- HAVE_PROJ4, 233
- HAVE_SQRT, 233
- HAVE_STD, 233
- HAVE_STDBOOL_H, 233
- HAVE_STDINT_H, 233
- HAVE_STDLIB_H, 233
- HAVE_STL, 233
- HAVE_STRING_H, 233
- HAVE_STRINGS_H, 233
- HAVE_STRSTR, 233
- HAVE_SYS_STAT_H, 233
- HAVE_SYS_TYPES_H, 233
- HAVE_UDUNITS, 233
- HAVE_UNISTD_H, 233
- LSTAT_FOLLOWS_SLASHED_SYMLINK, 233
- NETCDF_C_INCLUDE, 233
- NETCDF_CPP_INCLUDE, 233
- PACKAGE, 233
- PACKAGE_BUGREPORT, 233
- PACKAGE_NAME, 233
- PACKAGE_STRING, 233
- PACKAGE_TARNAME, 233
- PACKAGE_VERSION, 233
- STDC_HEADERS, 233
- VERSION, 233
- configFile
 - MetNoFimex::GribApiCDMWriter_ImplAbstract, 155
- const_iterator
 - KDTree::_Iterator, 63
 - KDTree::KDTree, 162
- const_pointer
 - KDTree::KDTree, 162
- const_reference
 - KDTree::KDTree, 162
- const_reverse_iterator
 - KDTree::KDTree, 162
- constConvertArrayType
 - MetNoFimex, 50
- convert
 - MetNoFimex::Units, 211
- convertArrayType
 - MetNoFimex, 50
- convertData
 - MetNoFimex::DataTypeChanger, 130
- convertDataType
 - MetNoFimex::Data, 120
 - MetNoFimex::DataImpl, 126
- count
 - KDTree::squared_difference_counted, 204
 - count_within_range
 - KDTree::KDTree, 169
- createData
 - MetNoFimex, 50, 51
- createDataSlice
 - MetNoFimex, 51
- DataImpl
 - MetNoFimex::DataImpl, 123
- datatype2string
 - MetNoFimex, 51
- DataTypeChanger
 - MetNoFimex::DataTypeChanger, 129
- DEBUG
 - MetNoFimex::Logger, 187
- DEFAULT_CONFIG
 - MetNoFelt::FeltParameters, 143
- defaultLogLevel
 - MetNoFimex, 51
- difference_type
 - KDTree::_Iterator, 63
 - KDTree::KDTree, 162
- DimVec
 - MetNoFimex::CDM, 82
- disconnect
 - KDTree::_Alloc_base::NoLeakAlloc, 58
- distance_type
 - KDTree::KDTree, 162
 - KDTree::squared_difference, 203
 - KDTree::squared_difference_counted, 204
- doxydoc.txt, 215
- duplicateArrayType
 - MetNoFimex, 51
- efficient_replace_and_optimise
 - KDTree::KDTree, 163
- empty
 - KDTree::KDTree, 164
- encloses
 - KDTree::_Region, 73
- end
 - KDTree::KDTree, 166
- epochSeconds2unitTime
 - MetNoFimex::TimeUnit, 209
- erase
 - KDTree::KDTree, 168
- erase_exact
 - KDTree::KDTree, 168
- ERROR
 - MetNoFimex::Logger, 187
- FATAL
 - MetNoFimex::Logger, 187
- Felt_Array

- MetNoFelt::Felt_Array, 132
- Felt_File
 - MetNoFelt::Felt_File, 137
- Felt_File_Error
 - MetNoFelt::Felt_File_Error, 140
- FeltCDMReader
 - MetNoFimex::FeltCDMReader, 141
- FeltParameters
 - MetNoFelt::FeltParameters, 142
- fimexTime2unitTime
 - MetNoFimex::TimeUnit, 209
- fimexTime2unitTimeX
 - MetNoFimex::TimeUnit, 209
- find
 - KDTree::KDTree, 168
- find_exact
 - KDTree::KDTree, 169
- find_nearest
 - KDTree::KDTree, 171
- find_nearest_if
 - KDTree::KDTree, 171
- find_within_range
 - KDTree::KDTree, 170
- find_within_range_iterative
 - KDTree::KDTree, 170
- findVariables
 - MetNoFimex::CDM, 83
- forcedLog
 - MetNoFimex::Logger, 187
- generateProjectionCoordinates
 - MetNoFimex::CDM, 88
- get
 - KDTree::_Alloc_base::NoLeakAlloc, 58
- get_allocator
 - KDTree::_Alloc_base, 56
 - KDTree::KDTree, 164
- get_raw_node
 - KDTree::_Iterator, 63
- getAttribute
 - MetNoFimex::CDM, 87, 88
 - MetNoFimex::NetCDF_CDMWriter, 192
- getAttributes
 - MetNoFimex::CDM, 87
- getAxisSteps
 - MetNoFimex::SpatialAxisSpec, 202
- getCDM
 - MetNoFimex::CDMReader, 108
 - MetNoFimex::FeltCDMReader, 141
- getData
 - MetNoFimex::CDMAttribute, 93
 - MetNoFimex::CDMReader, 108
 - MetNoFimex::CDMVariable, 114
- getDataHeader
 - MetNoFelt::Felt_Array, 133
- getDataPtr
 - MetNoFimex::Data, 117
 - MetNoFimex::DataImpl, 123
- getDataSlice
 - MetNoFelt::Felt_File, 137
 - MetNoFimex::CDMExtractor, 98
 - MetNoFimex::CDMInterpolator, 101
 - MetNoFimex::CDMReader, 108
 - MetNoFimex::CDMTimeInterpolator, 110
 - MetNoFimex::FeltCDMReader, 141
 - MetNoFimex::NcmlCDMReader, 190
 - MetNoFimex::NetCDF_CF10_CDMReader, 193
- getDataSliceFromMemory
 - MetNoFimex::CDMReader, 109
- getDataType
 - MetNoFimex::CDMAttribute, 93
 - MetNoFimex::CDMVariable, 113
 - MetNoFimex::Data, 120
 - MetNoFimex::DataImpl, 126
 - MetNoFimex::DataTypeChanger, 130
- getDatatype
 - MetNoFelt::Felt_Array, 133
- getDimension
 - MetNoFimex::CDM, 84, 85
- getDimensionName
 - MetNoFimex::NetCDF_CDMWriter, 191
- getDimensions
 - MetNoFimex::CDM, 87
- getFeltArray
 - MetNoFelt::Felt_File, 137
- getFeltLevelPairs
 - MetNoFelt::Felt_File, 138
- getFeltLevels
 - MetNoFelt::Felt_File, 138
- getFeltTimes
 - MetNoFelt::Felt_File, 138
- getFieldSize
 - MetNoFelt::Felt_Array, 135
- getFillValue
 - MetNoFelt::Felt_Array, 133
 - MetNoFimex::CDM, 88
- getGridParameters
 - MetNoFelt::Felt_Array, 133
 - MetNoFelt::Felt_File, 139
- getGridType
 - MetNoFelt::Felt_Array, 133
 - MetNoFelt::Felt_File, 139
- getHorizontalXAxis
 - MetNoFimex::CDM, 89
- getHorizontalYAxis
 - MetNoFimex::CDM, 89
- getHybridLevels

- MetNoFelt::Felt_File, 138
- getIdent19
 - MetNoFelt::Felt_Array, 134
- getIndex
 - MetNoFelt::Felt_Array, 135
- getIndexHeader
 - MetNoFelt::Felt_Array, 132
- getLatitudeLongitude
 - MetNoFimex::CDM, 90
- getLatitudeName
 - MetNoFimex::CDMInterpolator, 102
- getLength
 - MetNoFimex::CDMDimension, 96
- getLevelPairs
 - MetNoFelt::Felt_Array, 134
- getLevels
 - MetNoFelt::Felt_Array, 134
 - MetNoFimex::GribApiCDMWriter_ - ImplAbstract, 154
- getLevelType
 - MetNoFelt::Felt_Array, 133
- getLogger
 - MetNoFimex, 51
- getLongitudeName
 - MetNoFimex::CDMInterpolator, 103
- getName
 - MetNoFelt::Felt_Array, 133
 - MetNoFimex::CDMAttribute, 93
 - MetNoFimex::CDMDimension, 95
 - MetNoFimex::CDMNamedEntity, 105
 - MetNoFimex::CDMVariable, 113
- getNodePtr
 - MetNoFimex::GribApiCDMWriter_ - ImplAbstract, 154
- getNX
 - MetNoFelt::Felt_File, 138
- getNY
 - MetNoFelt::Felt_File, 139
- getParameterDatatype
 - MetNoFelt::FeltParameters, 143
- getParameterFillValue
 - MetNoFelt::FeltParameters, 143
- getParameterName
 - MetNoFelt::FeltParameters, 143
- getParameters
 - MetNoFelt::FeltParameters, 143
- getProjection
 - MetNoFimex::CDM, 89
- getProjectionAndAxesUnits
 - MetNoFimex::CDM, 89
- getProjString
 - MetNoFelt, 46
- getScaledData
 - MetNoFimex::CDMReader, 109
- getScaledDataSlice
 - MetNoFelt::Felt_File, 138
 - MetNoFimex::CDMReader, 108
- getScalingFactor
 - MetNoFelt::Felt_Array, 135
- getShape
 - MetNoFimex::CDMVariable, 113
- getSpatialVectorCounterpart
 - MetNoFimex::CDMVariable, 113
- getSpatialVectorDirection
 - MetNoFimex::CDMVariable, 113
- getStringValue
 - MetNoFimex::CDMAttribute, 93
- getTimeAxis
 - MetNoFimex::CDM, 90
- getTimeLevelSlice
 - MetNoFimex::TimeLevelDataSliceFetcher, 205
- getTimes
 - MetNoFelt::Felt_Array, 134
 - MetNoFimex::GribApiCDMWriter_ - ImplAbstract, 154
- getTimeSteps
 - MetNoFimex::TimeSpec, 207
- getUnitString
 - MetNoFimex::TimeSpec, 207
- getUnlimitedDim
 - MetNoFimex::CDM, 85
- getVariable
 - MetNoFimex::CDM, 82
- getVariableName
 - MetNoFimex::NetCDF_CDMWriter, 191
- getVariables
 - MetNoFimex::CDM, 87
- getVerticalAxis
 - MetNoFimex::CDM, 90
- getVerticalFeltType
 - MetNoFelt::Felt_Array, 135
- getX
 - MetNoFelt::Felt_Array, 135
- getXData
 - MetNoFelt::Felt_File, 139
- getXmlName
 - MetNoFimex, 52
- getXmlProp
 - MetNoFimex, 52
- getXPathObject
 - MetNoFimex::XMLDoc, 213
- getXSize
 - MetNoFimex::CachedVectorReprojection, 78
- getY
 - MetNoFelt::Felt_Array, 135
- getYData
 - MetNoFelt::Felt_File, 139

- getYSize
 - MetNoFimex::CachedVectorReprojection, [78](#)
- globalAttributeNS
 - MetNoFimex::CDM, [87](#)
- GribApiCDMWriter
 - MetNoFimex::GribApiCDMWriter, [147](#)
- GribApiCDMWriter_Impl1
 - MetNoFimex::GribApiCDMWriter_Impl1, [148](#)
- GribApiCDMWriter_Impl2
 - MetNoFimex::GribApiCDMWriter_Impl2, [150](#)
- GribApiCDMWriter_ImplAbstract
 - MetNoFimex::GribApiCDMWriter_ImplAbstract, [153](#)
- gribHandle
 - MetNoFimex::GribApiCDMWriter_ImplAbstract, [155](#)
- gribVersion
 - MetNoFimex::GribApiCDMWriter_ImplAbstract, [155](#)
- handleTypeScaleAndMissingData
 - MetNoFimex::GribApiCDMWriter_Impl1, [149](#)
 - MetNoFimex::GribApiCDMWriter_Impl2, [151](#)
 - MetNoFimex::GribApiCDMWriter_ImplAbstract, [154](#)
- handleUdUnitError
 - MetNoFimex, [52](#)
- hasData
 - MetNoFimex::CDMVariable, [114](#)
- hasDimension
 - MetNoFimex::CDM, [84](#)
- hasUnlimitedDim
 - MetNoFimex::CDM, [86](#)
- hasVariable
 - MetNoFimex::CDM, [83](#)
- HAVE_BOOST
 - config.h, [233](#)
- HAVE_BOOST_PROGRAM_OPTIONS
 - config.h, [233](#)
- HAVE_BOOST_REGEX
 - config.h, [233](#)
- HAVE_BOOST_UNIT_TEST_FRAMEWORK
 - config.h, [233](#)
- HAVE_CEIL
 - config.h, [233](#)
- HAVE_DLFCN_H
 - config.h, [233](#)
- HAVE_FMOD
 - config.h, [233](#)
- HAVE_GRIBAPI_H
 - config.h, [233](#)
- HAVE_INTTYPES_H
 - config.h, [233](#)
- HAVE_LIBM
 - config.h, [233](#)
- HAVE_LIBMIC
 - config.h, [233](#)
- HAVE_LOG10
 - config.h, [233](#)
- HAVE_MEMORY_H
 - config.h, [233](#)
- HAVE_MEMSET
 - config.h, [233](#)
- HAVE_MILIB_MILIB_H
 - config.h, [233](#)
- HAVE_NAMESPACES
 - config.h, [233](#)
- HAVE_NCFILE_FILEFORMAT
 - config.h, [233](#)
- HAVE_NETCDF
 - config.h, [233](#)
- HAVE_OPENMP
 - config.h, [233](#)
- HAVE_POW
 - config.h, [233](#)
- HAVE_PROJ4
 - config.h, [233](#)
- HAVE_SQRT
 - config.h, [233](#)
- HAVE_STD
 - config.h, [233](#)
- HAVE_STDBOOL_H
 - config.h, [233](#)
- HAVE_STDINT_H
 - config.h, [233](#)
- HAVE_STDLIB_H
 - config.h, [233](#)
- HAVE_STL
 - config.h, [233](#)
- HAVE_STRING_H
 - config.h, [233](#)
- HAVE_STRINGS_H
 - config.h, [233](#)
- HAVE_STRSTR
 - config.h, [233](#)
- HAVE_SYS_STAT_H
 - config.h, [233](#)
- HAVE_SYS_TYPES_H
 - config.h, [233](#)
- HAVE_UDUNITS
 - config.h, [233](#)
- HAVE_UNISTD_H
 - config.h, [233](#)
- hour

- MetNoFimex::FimexTime, 146
- include/fimex/CachedInterpolation.h, 216
- include/fimex/CachedVectorReprojection.h, 217
- include/fimex/CDM.h, 218
- include/fimex/CDMAAttribute.h, 219
- include/fimex/CDMconstants.h, 220
- include/fimex/CDMDataType.h, 221
- include/fimex/CDMDimension.h, 222
- include/fimex/CDMException.h, 223
- include/fimex/CDMExtractor.h, 224
- include/fimex/CDMInterpolator.h, 225
- include/fimex/CDMNamedEntity.h, 226
- include/fimex/CDMReader.h, 227
- include/fimex/CDMTimeInterpolator.h, 228
- include/fimex/CDMVariable.h, 229
- include/fimex/CDMWriter.h, 230
- include/fimex/config.h, 231
- include/fimex/Data.h, 234
- include/fimex/DataImpl.h, 235
- include/fimex/DataTypeChanger.h, 236
- include/fimex/Felt_Array.h, 237
- include/fimex/Felt_File.h, 238
- include/fimex/Felt_File_Error.h, 239
- include/fimex/FeltCDMReader.h, 240
- include/fimex/FeltParameters.h, 241
- include/fimex/GribApiCDMWriter.h, 242
- include/fimex/GribApiCDMWriter_Impl1.h, 243
- include/fimex/GribApiCDMWriter_Impl2.h, 244
- include/fimex/GribApiCDMWriter_ -
ImplAbstract.h, 245
- include/fimex/interpolation.h, 246
- include/fimex/Logger.h, 256
- include/fimex/NcmiCDMReader.h, 257
- include/fimex/NetCDF_CDMWriter.h, 258
- include/fimex/NetCDF_CF10_CDMReader.h, 259
- include/fimex/NetCDF_Utills.h, 260
- include/fimex/Null_CDMWriter.h, 261
- include/fimex/ReplaceStringObject.h, 262
- include/fimex/ReplaceStringTimeObject.h, 263
- include/fimex/SpatialAxisSpec.h, 264
- include/fimex/TimeLevelDataSliceFetcher.h, 265
- include/fimex/TimeSpec.h, 266
- include/fimex/TimeUnit.h, 267
- include/fimex/Units.h, 268
- include/fimex/Utils.h, 269
- include/fimex/XMLDoc.h, 270
- include/kdtree++/allocator.hpp, 271
- include/kdtree++/function.hpp, 272
- include/kdtree++/iterator.hpp, 273
- include/kdtree++/kdtree.hpp, 274
- include/kdtree++/node.hpp, 276
- include/kdtree++/region.hpp, 277
- index16toLevelPair
 - MetNoFelt, 46
- index16toTime
 - MetNoFelt, 46
- INFO
 - MetNoFimex::Logger, 187
- insert
 - KDTree::KDTree, 167, 168
- interpolateValues
 - MetNoFimex::CachedInterpolation, 76
- interpolation.h
 - mifi_3d_array_position, 249
 - mifi_bad2nanf, 250
 - MIFI_BICUBIC, 248
 - MIFI_BILINEAR, 248
 - MIFI_COORD_NN, 248
 - MIFI_COORD_NN_KD, 248
 - MIFI_DEBUG, 248
 - MIFI_ERROR, 248
 - mifi_get_values_bicubic_f, 250
 - mifi_get_values_bilinear_f, 250
 - mifi_get_values_f, 251
 - mifi_get_values_linear_f, 251
 - mifi_get_vector_reproject_matrix, 251
 - mifi_interpolate_d, 252
 - mifi_interpolate_f, 252
 - MIFI_LATITUDE, 248
 - MIFI_LONGITUDE, 249
 - mifi_nanf2bad, 253
 - MIFI_NEAREST_NEIGHBOR, 249
 - MIFI_OK, 249
 - mifi_points2position, 253
 - MIFI_PROJ_AXIS, 249
 - mifi_project_axes, 253
 - mifi_project_values, 254
 - MIFI_UNDEFINED_D, 249
 - MIFI_UNDEFINED_F, 249
 - MIFI_VECTOR_KEEP_SIZE, 249
 - mifi_vector_reproject_values_by_matrix_f,
254
 - mifi_vector_reproject_values_f, 255
 - MIFI_VECTOR_RESIZE, 249
 - PI, 249
- intersects_with
 - KDTree::_Region, 72, 73
- isEnabledFor
 - MetNoFimex::Logger, 187
- isSpatialVector
 - MetNoFimex::CDMVariable, 113
- isTime
 - MetNoFimex::Units, 212
- isUnlimited
 - MetNoFimex::CDMDimension, 96
- iterator
 - KDTree::_Iterator, 63

- KDTree::KDTree, 162
- iterator_category
 - KDTree::_Iterator, 63
- join
 - MetNoFimex, 52
- KDTree, 41
 - _S_accumulate_node_distance, 42
 - _S_node_compare, 42
 - _S_node_descend, 42
 - _S_node_distance, 42
 - _S_node_nearest, 43
 - KDTree::_Base_iterator, 60
 - KDTree::KDTree, 162, 163
 - operator!=, 43
 - operator==, 43, 44
- kdtree.hpp
 - KDTREE_LIB_VERSION, 275
 - KDTREE_VERSION, 275
- KDTree::_Alloc_base, 55
 - _Alloc_base, 56
 - _Base_ptr, 56
 - _M_allocate_node, 56
 - _M_construct_node, 56
 - _M_deallocate_node, 56
 - _M_destroy_node, 57
 - _M_node_allocator, 57
 - _Node_, 56
 - allocator_type, 56
 - get_allocator, 56
- KDTree::_Alloc_base::NoLeakAlloc, 58
 - ~NoLeakAlloc, 58
 - disconnect, 58
 - get, 58
 - NoLeakAlloc, 58
- KDTree::_Base_iterator, 59
 - _Base_const_ptr, 59
 - _Base_iterator, 59
 - _M_decrement, 60
 - _M_increment, 59
 - _M_node, 60
 - KDTree, 60
- KDTree::_Bracket_accessor, 61
 - operator(), 61
 - result_type, 61
- KDTree::_Iterator, 62
 - _Iterator, 63
 - _Link_const_type, 63
 - _Self, 63
 - const_iterator, 63
 - difference_type, 63
 - get_raw_node, 63
 - iterator, 63
 - iterator_category, 63
 - operator!=, 65
 - operator*, 63
 - operator++, 64
 - operator->, 64
 - operator-, 64
 - operator==, 65
 - pointer, 63
 - reference, 63
 - value_type, 63
- KDTree::_Node, 66
 - _Link_type, 66
 - _M_value, 66
 - _Node, 66
- KDTree::_Node_base, 67
 - _Base_const_ptr, 68
 - _Base_ptr, 68
 - _M_left, 68
 - _M_parent, 68
 - _M_right, 68
 - _Node_base, 68
 - _S_maximum, 68
 - _S_minimum, 68
- KDTree::_Node_compare, 70
 - _Node_compare, 70
 - operator(), 70
- KDTree::_Region, 71
 - _CenterPt, 72
 - _M_acc, 74
 - _M_cmp, 74
 - _M_high_bounds, 74
 - _M_low_bounds, 73
 - _Region, 72
 - encloses, 73
 - intersects_with, 72, 73
 - set_high_bound, 73
 - set_low_bound, 73
 - subvalue_type, 72
 - value_type, 72
- KDTree::always_true, 75
 - operator(), 75
- KDTree::KDTree, 156
 - ~KDTree, 163
 - _Base, 160
 - _Base_const_ptr, 160
 - _Base_ptr, 160
 - _Link_const_type, 162
 - _Link_type, 162
 - _M_acc, 185
 - _M_check_children, 172
 - _M_check_node, 172
 - _M_cmp, 185
 - _M_count, 185
 - _M_count_within_range, 177

- [_M_delete_node](#), 184
- [_M_dist](#), 186
- [_M_empty_initialise](#), 173
- [_M_erase](#), 174
- [_M_erase_subtree](#), 175
- [_M_find](#), 175
- [_M_find_exact](#), 176
- [_M_find_within_range](#), 177
- [_M_get_erase_replacement](#), 174
- [_M_get_j_max](#), 175
- [_M_get_j_min](#), 175
- [_M_get_leftmost](#), 179
- [_M_get_rightmost](#), 179
- [_M_get_root](#), 178
- [_M_header](#), 184
- [_M_insert](#), 174
- [_M_insert_left](#), 173
- [_M_insert_right](#), 173
- [_M_matches_node](#), 177
- [_M_matches_node_in_d](#), 176
- [_M_matches_node_in_other_ds](#), 176
- [_M_new_node](#), 184
- [_M_optimise](#), 178
- [_M_root](#), 184
- [_M_set_leftmost](#), 179
- [_M_set_rightmost](#), 180
- [_M_set_root](#), 179
- [_M_visit_within_range](#), 177
- [_Node_compare_](#), 162
- [_Region_](#), 162
- [_S_is_leaf](#), 183
- [_S_left](#), 181, 182
- [_S_maximum](#), 183
- [_S_minimum](#), 183
- [_S_parent](#), 180
- [_S_right](#), 182
- [_S_set_left](#), 181
- [_S_set_parent](#), 181
- [_S_set_right](#), 182
- [_S_value](#), 183
- [allocator_type](#), 160
- [begin](#), 166
- [check_tree](#), 172
- [clear](#), 164
- [const_iterator](#), 162
- [const_pointer](#), 162
- [const_reference](#), 162
- [const_reverse_iterator](#), 162
- [count_within_range](#), 169
- [difference_type](#), 162
- [distance_type](#), 162
- [efficient_replace_and_optimise](#), 163
- [empty](#), 164
- [end](#), 166
- [erase](#), 168
- [erase_exact](#), 168
- [find](#), 168
- [find_exact](#), 169
- [find_nearest](#), 171
- [find_nearest_if](#), 171
- [find_within_range](#), 170
- [find_within_range_iterative](#), 170
- [get_allocator](#), 164
- [insert](#), 167, 168
- [iterator](#), 162
- [KDTree](#), 162, 163
- [max_size](#), 164
- [operator=](#), 163
- [optimise](#), 172
- [optimize](#), 172
- [pointer](#), 162
- [rbegin](#), 166
- [reference](#), 162
- [rend](#), 167
- [reverse_iterator](#), 162
- [size](#), 164
- [size_type](#), 162
- [subvalue_type](#), 162
- [value_acc](#), 165
- [value_comp](#), 165
- [value_distance](#), 165, 166
- [value_type](#), 162
- [visit_within_range](#), 169, 170
- [KDTree::squared_difference](#), 203
 - [distance_type](#), 203
 - [operator\(\)](#), 203
- [KDTree::squared_difference_counted](#), 204
 - [count](#), 204
 - [distance_type](#), 204
 - [operator\(\)](#), 204
 - [reset](#), 204
 - [squared_difference_counted](#), 204
- [KDTREE_LIB_VERSION](#)
 - [kdtree.hpp](#), 275
- [KDTREE_VERSION](#)
 - [kdtree.hpp](#), 275
- [listFeltArrays](#)
 - [MetNoFelt::Felt_File](#), 138
- [LOG4FIMEX](#)
 - [Logger.h](#), 256
- [Logger](#)
 - [MetNoFimex::Logger](#), 187
- [logger](#)
 - [MetNoFimex::GribApiCDMWriter_](#)
 - [ImplAbstract](#), 155
- [Logger.h](#)
 - [LOG4FIMEX](#), 256

- LoggerPtr
 - MetNoFimex, 49
- LogLevel
 - MetNoFimex::Logger, 187
- LSTAT_FOLLOWS_SLASHED_SYMLINK
 - config.h, 233
- max_size
 - KDTree::KDTree, 164
- mday
 - MetNoFimex::FimexTime, 146
- MetNoFelt, 45
 - ANY_ARRAY, 46
 - ANY_ARRAY20, 46
 - ANY_VALUE, 46
 - getProjString, 46
 - index16toLevelPair, 46
 - index16toTime, 46
 - ShortPairMap, 45
 - ShortPairSet, 45
 - UNDEFINED, 46
- MetNoFelt::Felt_Array, 131
 - ~Felt_Array, 132
 - addIdent19, 134
 - addInformationByIndex, 132
 - Felt_Array, 132
 - getDataHeader, 133
 - getDatatype, 133
 - getFieldSize, 135
 - getFillValue, 133
 - getGridParameters, 133
 - getGridType, 133
 - getIdent19, 134
 - getIndex, 135
 - getIndexHeader, 132
 - getLevelPairs, 134
 - getLevels, 134
 - getLevelType, 133
 - getName, 133
 - getScalingFactor, 135
 - getTimes, 134
 - getVerticalFeltType, 135
 - getX, 135
 - getY, 135
 - setDataHeader, 132
 - setFillValue, 133
 - setGridParameters, 133
 - setGridType, 133
- MetNoFelt::Felt_File, 136
 - ~Felt_File, 137
 - Felt_File, 137
 - getDataSlice, 137
 - getFeltArray, 137
 - getFeltLevelPairs, 138
 - getFeltLevels, 138
 - getFeltTimes, 138
 - getGridParameters, 139
 - getGridType, 139
 - getHybridLevels, 138
 - getNX, 138
 - getNY, 139
 - getScaledDataSlice, 138
 - getXData, 139
 - getYData, 139
 - listFeltArrays, 138
- MetNoFelt::Felt_File_Error, 140
 - ~Felt_File_Error, 140
 - Felt_File_Error, 140
 - what, 140
- MetNoFelt::FeltParameters, 142
 - ~FeltParameters, 142
 - DEFAULT_CONFIG, 143
 - FeltParameters, 142
 - getParameterDatatype, 143
 - getParameterFillValue, 143
 - getParameterName, 143
 - getParameters, 143
- MetNoFelt::ShortPairLess, 200
 - operator(), 200
- MetNoFimex, 47
 - attributesToProjString, 50
 - CDM_CHAR, 49
 - CDM_DOUBLE, 49
 - CDM_FLOAT, 49
 - CDM_INT, 49
 - CDM_NAT, 49
 - CDM_SHORT, 49
 - CDM_STRING, 49
 - CDMDataType, 49
 - cdmDataType2ncType, 50
 - constConvertArrayType, 50
 - convertArrayType, 50
 - createData, 50, 51
 - createDataSlice, 51
 - datatype2string, 51
 - defaultLogLevel, 51
 - duplicateArrayType, 51
 - getLogger, 51
 - getXmlName, 52
 - getXmlProp, 52
 - handleUdUnitError, 52
 - join, 52
 - LoggerPtr, 49
 - ncType2cdmDataType, 52
 - ncValues2Data, 52
 - operator<<, 52
 - projStringToAttributes, 53
 - recursiveCopyMultiDimData, 53

- round, 53
- string2datatype, 53
- string2FimexTime, 53
- string2lowerCase, 53
- string2type, 53
- tokenize, 53
- tokenizeDotted, 54
- trim, 54
- type2string, 54
- XPathObjPtr, 49
- MetNoFimex::CachedInterpolation, 76
 - ~CachedInterpolation, 76
 - CachedInterpolation, 76
 - interpolateValues, 76
- MetNoFimex::CachedVectorReprojection, 78
 - ~CachedVectorReprojection, 78
 - CachedVectorReprojection, 78
 - getXSize, 78
 - getYSize, 78
 - reprojectValues, 78
- MetNoFimex::CDM, 79
 - ~CDM, 82
 - addAttribute, 86
 - addDimension, 84
 - addOrReplaceAttribute, 86
 - addVariable, 82
 - AttrVec, 82
 - CDM, 82
 - checkVariableAttribute, 84
 - DimVec, 82
 - findVariables, 83
 - generateProjectionCoordinates, 88
 - getAttribute, 87, 88
 - getAttributes, 87
 - getDimension, 84, 85
 - getDimensions, 87
 - getFillValue, 88
 - getHorizontalXAxis, 89
 - getHorizontalYAxis, 89
 - getLatitudeLongitude, 90
 - getProjection, 89
 - getProjectionAndAxesUnits, 89
 - getTimeAxis, 90
 - getUnlimitedDim, 85
 - getVariable, 82
 - getVariables, 87
 - getVerticalAxis, 90
 - globalAttributeNS, 87
 - hasDimension, 84
 - hasUnlimitedDim, 86
 - hasVariable, 83
 - removeAttribute, 86
 - removeDimension, 85
 - removeVariable, 84
 - renameDimension, 85
 - renameVariable, 83
 - StrAttrVecMap, 82
 - testDimensionInUse, 85
 - toXMLStream, 86
 - VarVec, 82
- MetNoFimex::CDMAttribute, 91
 - ~CDMAttribute, 93
 - CDMAttribute, 92, 93
 - getData, 93
 - getDataType, 93
 - getName, 93
 - getStringValue, 93
 - setData, 93
 - setName, 93
 - toXMLStream, 93
- MetNoFimex::CDMDimension, 95
 - ~CDMDimension, 95
 - CDMDimension, 95
 - getLength, 96
 - getName, 95
 - isUnlimited, 96
 - setLength, 96
 - setName, 95
 - setUnlimited, 96
 - toXMLStream, 96
- MetNoFimex::CDMException, 97
 - ~CDMException, 97
 - CDMException, 97
 - operator=, 97
 - what, 97
- MetNoFimex::CDMExtractor, 98
 - ~CDMExtractor, 98
 - CDMExtractor, 98
 - changeDataType, 100
 - getDataSlice, 98
 - reduceDimension, 99
 - reduceDimensionStartEnd, 99
 - removeVariable, 99
- MetNoFimex::CDMInterpolator, 101
 - ~CDMInterpolator, 101
 - CDMInterpolator, 101
 - changeProjection, 101, 102
 - getDataSlice, 101
 - getLatitudeName, 102
 - getLongitudeName, 103
 - setLatitudeName, 102
 - setLongitudeName, 102
- MetNoFimex::CDMNameCompare, 104
 - operator(), 104
- MetNoFimex::CDMNamedEntity, 105
 - ~CDMNamedEntity, 105
 - getName, 105
- MetNoFimex::CDMNameEqual, 106

- ~CDMNameEqual, 106
- CDMNameEqual, 106
- operator(), 106
- MetNoFimex::CDMReader, 107
 - ~CDMReader, 108
 - cdm, 109
 - CDMReader, 108
 - getCDM, 108
 - getData, 108
 - getDataSlice, 108
 - getDataSliceFromMemory, 109
 - getScaledData, 109
 - getScaledDataSlice, 108
- MetNoFimex::CDMTimeInterpolator, 110
 - ~CDMTimeInterpolator, 110
 - CDMTimeInterpolator, 110
 - changeTimeAxis, 110
 - getDataSlice, 110
- MetNoFimex::CDMVariable, 112
 - ~CDMVariable, 113
 - CDMVariable, 113
 - checkDimension, 113
 - getData, 114
 - getDataType, 113
 - getName, 113
 - getShape, 113
 - getSpatialVectorCounterpart, 113
 - getSpatialVectorDirection, 113
 - hasData, 114
 - isSpatialVector, 113
 - setAsSpatialVector, 113
 - setData, 114
 - setDataType, 113
 - setName, 113
 - setShape, 113
 - toXMLStream, 114
- MetNoFimex::CDMWriter, 115
 - ~CDMWriter, 115
 - cdmReader, 115
 - CDMWriter, 115
 - outputFile, 115
- MetNoFimex::Data, 116
 - ~Data, 117
 - asChar, 118
 - asConstChar, 118
 - asConstDouble, 119
 - asConstFloat, 118
 - asConstInt, 118
 - asConstShort, 118
 - asDouble, 119
 - asFloat, 119
 - asInt, 118
 - asShort, 118
 - asString, 119
 - bytes_for_one, 117
 - convertDataType, 120
 - getDataPtr, 117
 - getDataType, 120
 - setValue, 119
 - setValues, 119
 - size, 117
 - slice, 119
 - toStream, 117
- MetNoFimex::DataImpl, 121
 - ~DataImpl, 123
 - as, 123, 124
 - asBase, 123
 - asChar, 124
 - asConstChar, 124
 - asConstDouble, 125
 - asConstFloat, 124
 - asConstInt, 124
 - asConstShort, 124
 - asDouble, 125
 - asFloat, 125
 - asInt, 124
 - asShort, 124
 - asString, 125
 - bytes_for_one, 123
 - convertDataType, 126
 - DataImpl, 123
 - getDataPtr, 123
 - getDataType, 126
 - setValue, 125
 - setValues, 125–128
 - size, 123
 - slice, 126
 - toStream, 123
- MetNoFimex::DataTypeChanger, 129
 - ~DataTypeChanger, 129
 - convertData, 130
 - DataTypeChanger, 129
 - getDataType, 130
- MetNoFimex::FeltCDMReader, 141
 - ~FeltCDMReader, 141
 - FeltCDMReader, 141
 - getCDM, 141
 - getDataSlice, 141
- MetNoFimex::FimexTime, 144
 - hour, 146
 - mday, 146
 - minute, 145
 - month, 146
 - msecond, 145
 - operator!=, 145
 - operator<, 145
 - operator<=, 145
 - operator>, 145

- operator>=, 145
- operator==, 145
- second, 145
- year, 146
- MetNoFimex::GribApiCDMWriter, 147
 - ~GribApiCDMWriter, 147
 - GribApiCDMWriter, 147
- MetNoFimex::GribApiCDMWriter_Impl1, 148
 - ~GribApiCDMWriter_Impl1, 148
 - GribApiCDMWriter_Impl1, 148
 - handleTypeScaleAndMissingData, 149
 - setLevel, 149
 - setParameter, 148
 - setProjection, 148
- MetNoFimex::GribApiCDMWriter_Impl2, 150
 - ~GribApiCDMWriter_Impl2, 150
 - GribApiCDMWriter_Impl2, 150
 - handleTypeScaleAndMissingData, 151
 - setLevel, 151
 - setParameter, 150
 - setProjection, 150
- MetNoFimex::GribApiCDMWriter_ImplAbstract, 152
 - ~GribApiCDMWriter_ImplAbstract, 153
 - cdmReader, 155
 - configFile, 155
 - getLevels, 154
 - getNodePtr, 154
 - getTimes, 154
 - GribApiCDMWriter_ImplAbstract, 153
 - gribHandle, 155
 - gribVersion, 155
 - handleTypeScaleAndMissingData, 154
 - logger, 155
 - outputFile, 155
 - run, 153
 - setData, 153
 - setGlobalAttributes, 153
 - setLevel, 154
 - setParameter, 153
 - setProjection, 153
 - setTime, 153
 - writeGribHandleToFile, 154
 - xmlConfig, 155
- MetNoFimex::Logger, 187
 - ~Logger, 187
 - DEBUG, 187
 - ERROR, 187
 - FATAL, 187
 - forcedLog, 187
 - INFO, 187
 - isEnabledFor, 187
 - Logger, 187
 - LogLevel, 187
 - OFF, 187
 - WARN, 187
- MetNoFimex::NcmlCDMReader, 189
 - ~NcmlCDMReader, 190
 - getDataSlice, 190
 - NcmlCDMReader, 189
- MetNoFimex::NetCDF_CDMWriter, 191
 - ~NetCDF_CDMWriter, 191
 - getAttribute, 192
 - getDimensionName, 191
 - getVariableName, 191
 - NetCDF_CDMWriter, 191
- MetNoFimex::NetCDF_CF10_CDMReader, 193
 - ~NetCDF_CF10_CDMReader, 193
 - getDataSlice, 193
 - NetCDF_CF10_CDMReader, 193
- MetNoFimex::Null_CDMWriter, 195
 - ~Null_CDMWriter, 195
 - Null_CDMWriter, 195
- MetNoFimex::ReplaceStringObject, 196
 - ~ReplaceStringObject, 196
 - put, 196
 - setFormatString, 196
 - setFormatStringAndOptions, 196
- MetNoFimex::ReplaceStringTimeObject, 198
 - ~ReplaceStringTimeObject, 198
 - operator<<, 199
 - put, 198
 - ReplaceStringTimeObject, 198
 - setFormatString, 198
 - setFormatStringAndOptions, 199
- MetNoFimex::SpatialAxisSpec, 201
 - ~SpatialAxisSpec, 202
 - getAxisSteps, 202
 - requireStartEnd, 202
 - setStartEnd, 202
 - SpatialAxisSpec, 201
- MetNoFimex::TimeLevelDataSliceFetcher, 205
 - ~TimeLevelDataSliceFetcher, 205
 - getTimeLevelSlice, 205
 - TimeLevelDataSliceFetcher, 205
- MetNoFimex::TimeSpec, 206
 - ~TimeSpec, 207
 - getTimeSteps, 207
 - getUnitString, 207
 - TimeSpec, 206
- MetNoFimex::TimeUnit, 208
 - ~TimeUnit, 209
 - epochSeconds2unitTime, 209
 - fimexTime2unitTime, 209
 - fimexTime2unitTimeX, 209
 - TimeUnit, 208
 - unitTime2epochSeconds, 209
 - unitTime2fimexTime, 209

- MetNoFimex::UnitException, 210
 - UnitException, 210
- MetNoFimex::Units, 211
 - ~Units, 211
 - areConvertible, 211
 - convert, 211
 - isTime, 212
 - operator=, 211
 - Units, 211
- MetNoFimex::XMLDoc, 213
 - ~XMLDoc, 213
 - getXPathObject, 213
 - registerNamespace, 214
 - XMLDoc, 213
- mifi_3d_array_position
 - interpolation.h, 249
- mifi_bad2nanf
 - interpolation.h, 250
- MIFI_BICUBIC
 - interpolation.h, 248
- MIFI_BILINEAR
 - interpolation.h, 248
- MIFI_COORD_NN
 - interpolation.h, 248
- MIFI_COORD_NN_KD
 - interpolation.h, 248
- MIFI_DEBUG
 - interpolation.h, 248
- MIFI_EARTH_RADIUS_M
 - CDMconstants.h, 220
- MIFI_ERROR
 - interpolation.h, 248
- mifi_get_values_bicubic_f
 - interpolation.h, 250
- mifi_get_values_bilinear_f
 - interpolation.h, 250
- mifi_get_values_f
 - interpolation.h, 251
- mifi_get_values_linear_f
 - interpolation.h, 251
- mifi_get_vector_reproject_matrix
 - interpolation.h, 251
- mifi_interpolate_d
 - interpolation.h, 252
- mifi_interpolate_f
 - interpolation.h, 252
- MIFI_LATITUDE
 - interpolation.h, 248
- MIFI_LONGITUDE
 - interpolation.h, 249
- mifi_nanf2bad
 - interpolation.h, 253
- MIFI_NEAREST_NEIGHBOR
 - interpolation.h, 249
- MIFI_OK
 - interpolation.h, 249
- mifi_points2position
 - interpolation.h, 253
- MIFI_PROJ_AXIS
 - interpolation.h, 249
- mifi_project_axes
 - interpolation.h, 253
- mifi_project_values
 - interpolation.h, 254
- MIFI_UNDEFINED_D
 - interpolation.h, 249
- MIFI_UNDEFINED_F
 - interpolation.h, 249
- MIFI_VECTOR_KEEP_SIZE
 - interpolation.h, 249
- mifi_vector_reproject_values_by_matrix_f
 - interpolation.h, 254
- mifi_vector_reproject_values_f
 - interpolation.h, 255
- MIFI_VECTOR_RESIZE
 - interpolation.h, 249
- minute
 - MetNoFimex::FimexTime, 145
- month
 - MetNoFimex::FimexTime, 146
- msecond
 - MetNoFimex::FimexTime, 145
- NcmlCDMReader
 - MetNoFimex::NcmlCDMReader, 189
- ncType2cdmDataType
 - MetNoFimex, 52
- ncValues2Data
 - MetNoFimex, 52
- NETCDF_C_INCLUDE
 - config.h, 233
- NetCDF_CDMWriter
 - MetNoFimex::NetCDF_CDMWriter, 191
- NetCDF_CF10_CDMReader
 - MetNoFimex::NetCDF_CF10_CDMReader, 193
- NETCDF_CPP_INCLUDE
 - config.h, 233
- NoLeakAlloc
 - KDTree::_Alloc_base::NoLeakAlloc, 58
- Null_CDMWriter
 - MetNoFimex::Null_CDMWriter, 195
- OFF
 - MetNoFimex::Logger, 187
- operator!=
 - KDTree, 43
 - KDTree::_Iterator, 65

- MetNoFimex::FimexTime, 145
- operator<
 - MetNoFimex::FimexTime, 145
- operator<<
 - MetNoFimex, 52
 - MetNoFimex::ReplaceStringTimeObject, 199
- operator<=
 - MetNoFimex::FimexTime, 145
- operator>
 - MetNoFimex::FimexTime, 145
- operator>=
 - MetNoFimex::FimexTime, 145
- operator*
 - KDTree::_Iterator, 63
- operator()
 - KDTree::_Bracket_accessor, 61
 - KDTree::_Node_compare, 70
 - KDTree::always_true, 75
 - KDTree::squared_difference, 203
 - KDTree::squared_difference_counted, 204
 - MetNoFelt::ShortPairLess, 200
 - MetNoFimex::CDMNameCompare, 104
 - MetNoFimex::CDMNameEqual, 106
- operator++
 - KDTree::_Iterator, 64
- operator->
 - KDTree::_Iterator, 64
- operator-
 - KDTree::_Iterator, 64
- operator=
 - KDTree::KDTree, 163
 - MetNoFimex::CDMException, 97
 - MetNoFimex::Units, 211
- operator==
 - KDTree, 43, 44
 - KDTree::_Iterator, 65
 - MetNoFimex::FimexTime, 145
- optimise
 - KDTree::KDTree, 172
- optimize
 - KDTree::KDTree, 172
- outputFile
 - MetNoFimex::CDMWriter, 115
 - MetNoFimex::GribApiCDMWriter_ImplAbstract, 155
- PACKAGE
 - config.h, 233
- PACKAGE_BUGREPORT
 - config.h, 233
- PACKAGE_NAME
 - config.h, 233
- PACKAGE_STRING
 - config.h, 233
- PACKAGE_TARNAME
 - config.h, 233
- PACKAGE_VERSION
 - config.h, 233
- PI
 - interpolation.h, 249
- pointer
 - KDTree::_Iterator, 63
 - KDTree::KDTree, 162
- projStringToAttributes
 - MetNoFimex, 53
- put
 - MetNoFimex::ReplaceStringObject, 196
 - MetNoFimex::ReplaceStringTimeObject, 198
- rbegin
 - KDTree::KDTree, 166
- recursiveCopyMultiDimData
 - MetNoFimex, 53
- reduceDimension
 - MetNoFimex::CDMExtractor, 99
- reduceDimensionStartEnd
 - MetNoFimex::CDMExtractor, 99
- reference
 - KDTree::_Iterator, 63
 - KDTree::KDTree, 162
- registerNamespace
 - MetNoFimex::XMLDoc, 214
- removeAttribute
 - MetNoFimex::CDM, 86
- removeDimension
 - MetNoFimex::CDM, 85
- removeVariable
 - MetNoFimex::CDM, 84
 - MetNoFimex::CDMExtractor, 99
- renameDimension
 - MetNoFimex::CDM, 85
- renameVariable
 - MetNoFimex::CDM, 83
- rend
 - KDTree::KDTree, 167
- ReplaceStringTimeObject
 - MetNoFimex::ReplaceStringTimeObject, 198
- reprojectValues
 - MetNoFimex::CachedVectorReprojection, 78
- requireStartEnd
 - MetNoFimex::SpatialAxisSpec, 202
- reset
 - KDTree::squared_difference_counted, 204
- result_type
 - KDTree::_Bracket_accessor, 61
- reverse_iterator
 - KDTree::KDTree, 162
- round

- MetNoFimex, 53
- run
 - MetNoFimex::GribApiCDMWriter_ImplAbstract, 153
- second
 - MetNoFimex::FimexTime, 145
- set_high_bound
 - KDTree::_Region, 73
- set_low_bound
 - KDTree::_Region, 73
- setAsSpatialVector
 - MetNoFimex::CDMVariable, 113
- setData
 - MetNoFimex::CDMAttribute, 93
 - MetNoFimex::CDMVariable, 114
 - MetNoFimex::GribApiCDMWriter_ImplAbstract, 153
- setDataHeader
 - MetNoFelt::Felt_Array, 132
- setDataType
 - MetNoFimex::CDMVariable, 113
- setFillValue
 - MetNoFelt::Felt_Array, 133
- setFormatString
 - MetNoFimex::ReplaceStringObject, 196
 - MetNoFimex::ReplaceStringTimeObject, 198
- setFormatStringAndOptions
 - MetNoFimex::ReplaceStringObject, 196
 - MetNoFimex::ReplaceStringTimeObject, 199
- setGlobalAttributes
 - MetNoFimex::GribApiCDMWriter_ImplAbstract, 153
- setGridParameters
 - MetNoFelt::Felt_Array, 133
- setGridType
 - MetNoFelt::Felt_Array, 133
- setLatitudeName
 - MetNoFimex::CDMInterpolator, 102
- setLength
 - MetNoFimex::CDMDimension, 96
- setLevel
 - MetNoFimex::GribApiCDMWriter_Impl1, 149
 - MetNoFimex::GribApiCDMWriter_Impl2, 151
 - MetNoFimex::GribApiCDMWriter_ImplAbstract, 154
- setLongitudeName
 - MetNoFimex::CDMInterpolator, 102
- setName
 - MetNoFimex::CDMAttribute, 93
 - MetNoFimex::CDMDimension, 95
 - MetNoFimex::CDMVariable, 113
- setParameter
 - MetNoFimex::GribApiCDMWriter_Impl1, 148
 - MetNoFimex::GribApiCDMWriter_Impl2, 150
 - MetNoFimex::GribApiCDMWriter_ImplAbstract, 153
- setProjection
 - MetNoFimex::GribApiCDMWriter_Impl1, 148
 - MetNoFimex::GribApiCDMWriter_Impl2, 150
 - MetNoFimex::GribApiCDMWriter_ImplAbstract, 153
- setShape
 - MetNoFimex::CDMVariable, 113
- setStartEnd
 - MetNoFimex::SpatialAxisSpec, 202
- setTime
 - MetNoFimex::GribApiCDMWriter_ImplAbstract, 153
- setUnlimited
 - MetNoFimex::CDMDimension, 96
- setValue
 - MetNoFimex::Data, 119
 - MetNoFimex::DataImpl, 125
- setValues
 - MetNoFimex::Data, 119
 - MetNoFimex::DataImpl, 125–128
- ShortPairMap
 - MetNoFelt, 45
- ShortPairSet
 - MetNoFelt, 45
- size
 - KDTree::KDTree, 164
 - MetNoFimex::Data, 117
 - MetNoFimex::DataImpl, 123
- size_type
 - KDTree::KDTree, 162
- slice
 - MetNoFimex::Data, 119
 - MetNoFimex::DataImpl, 126
- SpatialAxisSpec
 - MetNoFimex::SpatialAxisSpec, 201
- squared_difference_counted
 - KDTree::squared_difference_counted, 204
- STDC_HEADERS
 - config.h, 233
- StrAttrVecMap
 - MetNoFimex::CDM, 82
- string2datatype
 - MetNoFimex, 53
- string2FimexTime
 - MetNoFimex, 53

- string2lowerCase
 - MetNoFimex, [53](#)
- string2type
 - MetNoFimex, [53](#)
- subvalue_type
 - KDTree::_Region, [72](#)
 - KDTree::KDTree, [162](#)
- testDimensionInUse
 - MetNoFimex::CDM, [85](#)
- TimeLevelDataSliceFetcher
 - MetNoFimex::TimeLevelDataSliceFetcher, [205](#)
- TimeSpec
 - MetNoFimex::TimeSpec, [206](#)
- TimeUnit
 - MetNoFimex::TimeUnit, [208](#)
- tokenize
 - MetNoFimex, [53](#)
- tokenizeDotted
 - MetNoFimex, [54](#)
- toStream
 - MetNoFimex::Data, [117](#)
 - MetNoFimex::DataImpl, [123](#)
- toXMLStream
 - MetNoFimex::CDM, [86](#)
 - MetNoFimex::CDMAttribute, [93](#)
 - MetNoFimex::CDMDimension, [96](#)
 - MetNoFimex::CDMVariable, [114](#)
- trim
 - MetNoFimex, [54](#)
- type2string
 - MetNoFimex, [54](#)
- UNDEFINED
 - MetNoFelt, [46](#)
- UnitException
 - MetNoFimex::UnitException, [210](#)
- Units
 - MetNoFimex::Units, [211](#)
- unitTime2epochSeconds
 - MetNoFimex::TimeUnit, [209](#)
- unitTime2fimexTime
 - MetNoFimex::TimeUnit, [209](#)
- value_acc
 - KDTree::KDTree, [165](#)
- value_comp
 - KDTree::KDTree, [165](#)
- value_distance
 - KDTree::KDTree, [165](#), [166](#)
- value_type
 - KDTree::_Iterator, [63](#)
 - KDTree::_Region, [72](#)
 - KDTree::KDTree, [162](#)
- VarVec
 - MetNoFimex::CDM, [82](#)
- VERSION
 - config.h, [233](#)
- visit_within_range
 - KDTree::KDTree, [169](#), [170](#)
- WARN
 - MetNoFimex::Logger, [187](#)
- what
 - MetNoFelt::Felt_File_Error, [140](#)
 - MetNoFimex::CDMException, [97](#)
- writeGribHandleToFile
 - MetNoFimex::GribApiCDMWriter_-ImplAbstract, [154](#)
- xmlConfig
 - MetNoFimex::GribApiCDMWriter_-ImplAbstract, [155](#)
- XMLDoc
 - MetNoFimex::XMLDoc, [213](#)
- XPathObjPtr
 - MetNoFimex, [49](#)
- year
 - MetNoFimex::FimexTime, [146](#)