

SmartMet Server GRID FILES

Finnish Meteorological Institute

2021-11-17

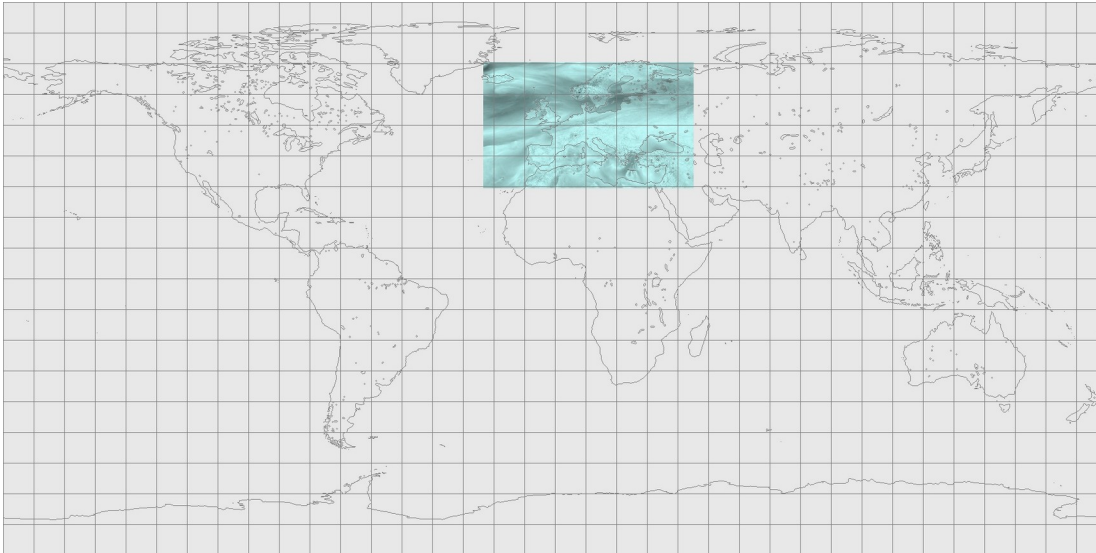
Table of Contents

1 GRID files.....	3
1.1 Introduction.....	3
1.2 File formats.....	3
2 FMI Identifiers.....	5
2.1 Introduction.....	5
2.2 FMI parameter identifiers.....	5
2.3 FMI level type identifiers.....	6
2.4 FMI geometry identifiers.....	6
3 GRIB Identifiers.....	7
3.1 Introduction.....	7
3.2 GRIB parameter identifiers.....	7
3.2.1 GRIB1 / GRIB2 field values => GRIB parameter identifiers.....	7
3.2.2 GRIB parameter identifiers => FMI parameter identifiers.....	8
3.3 GRIB level type identifiers.....	8
3.3.1 GRIB1 / GRIB2 field values => FMI level identifiers.....	8
4 NetCDF Identifiers.....	10
4.1 Introduction.....	10
4.2 NetCDF parameter identifiers.....	10
4.2.1 NetCDF parameter identifiers => FMI parameter identifiers.....	10
4.3 NetCDF level identifiers.....	10
5 Newbase Identifiers.....	11
5.1 Introduction.....	11
5.2 Newbase parameter identifiers.....	11
5.2.1 Newbase parameter identifiers => FMI parameter identifiers.....	11
5.3 Newbase level type identifiers.....	12
6 Summary.....	12

1 GRID files

1.1 Introduction

Probably the easiest way to understand a grid is to think it as a two-dimensional matrix of points. Each point has a 1) geographical location and 2) a value. These values can represent for example weather forecast values (temperature, pressure, etc.). The figure below shows how grid values can be drawn into the world map.



1.2 File formats

Grid information can be stored into many kinds of grid files. The most well-known grid file formats are GRIB1, GRIB2 and NetCDF.

It's usually quite easy to get actual grid values out of different grid files. However, it's quite challenging to identify this grid data in such way that it can be queried in a general way. That's because different file formats identify grid data (= so called "parameters") in different ways. For example, we might have different grid files (in different formats) that contain same temperature information, but each format identifies this temperature information differently. Without any additional mapping it is difficult to know that these files contain the same information. That's why it is difficult to query information over different file types.

For example, GRIB files use multiple (header) fields in order to identify grid data. On the other hand, different producers identify the same data in different ways.

In a GRIB1 file the "temperature" parameter can be identified at least with the following field combinations:

```
table2Version=1,indicatorOfParameter=11  
table2Version=160,indicatorOfParameter=130  
table2Version=128,indicatorOfParameter=130  
table2Version=200,indicatorOfParameter=11
```

In a GRIB2 file, the same "temperature" parameter is identified with different fields:

```
discipline=0,parameterCategory=0,parameterNumber=0
```

Usually these field combinations are mapped into single "grib identifier". For example, the grib identifier for the "temperature" parameter is 130.

In a NetCDF file, the "temperature" parameter is usually identified with the "standard_name" field:

```
standard_name=air_temperature
```

However, this is true only with the newest NetCDF files that follow this "standard naming". Technically the NetCDF format does not require that, which makes the systematical identification of grid parameters almost impossible. If parameter names can be expressed in non-standard ways, then we can have hundreds of different names for the same parameter. This is a nightmare from the query point of view.

In a QueryData file (= FMI's internal format) parameters are identified with Newbase names or numbers. For example, the Newbase name of the "temperature" parameter is "temperature" and the number is 4.

If we want to query data over different kinds of grid files, we should map all file format specific identifiers into identifiers that do not depend on file formats. In this way we can query data without knowing details of different file formats. In practice, we have to map file format specific 1) parameter identifiers and 2) level type identifiers into format independent parameter identifier and level type identifiers. After that we can query information over different file formats. On the other hand, this means that we can easily add new file formats into the system, but still query data in the same way.

In our implementation we call these format independent identifiers as FMI parameter identifiers and FMI level identifiers (for historical reasons). All FMI parameter identifiers and FMI level identifiers have a unique name and a numeric id.

2 FMI Identifiers

2.1 Introduction

FMI identifiers are unique grid parameter identifiers that have no relationship to actual grid file formats. These identifiers are used for querying grid files in a format independent way. This requires that file format dependent parameter and level identifiers can be mapped into FMI parameter and level identifiers. This is done in different mappings files.

2.2 FMI parameter identifiers

All FMI parameter identifiers have unique names (= fmi-name) and numeric values (= fmi-id). FMI parameter names are used in order to describe the most essential information of the grid data in a such way that it can be easily remembered. That's because this parameter name is usually used for data queries.

A typical FMI parameter name consists of the short name of the variable/quantity and abbreviation of the units of the actual data. Here are some examples:

ID	NAME	DESCRIPTION
153	T-K	Temperature in Kelvins
154	T-C	Temperature in Celsius
163	RH-PRCNT	Relative Humidity in percents
431	TSEA-K	Sea Temperature in Kelvins
432	TSEA-C	Sea Temperature in Celsius

There should be different FMI parameter names for different variables/quantities and different units. For example, there should be different parameter names for air temperature, ground temperature and sea temperature. On the other hand, there should be different parameter names for air temperatures that are expressed in different units (Kelvin, Celsius, Fahrenheit). However, the FMI parameter name should not usually contain any level information. Typically the level information is seen as a part of location information (latitude, longitude, level/height => x, y, z) in data queries. The point is that it does not make sense to define different parameter names for all possible levels.

Usually the actual grid data and its format dependent parameter identifiers (grib-id, netcdf-name, newbase-id, etc.) should be mapped to FMI identifiers as accurately as possible. For example, if the original temperature data is expressed in Kelvins, then it should be also mapped to a FMI parameter that is used for Kelvins (= T-K), not to a parameter that is used for Celsius (= T-C).

In spite of that the SmartMet Server can make unit conversions during the queries, these conversions should be minimized. However, sometimes it makes sense to convert some units into the same unit. For example, if the original data is in "cm" (= centimeter) or "mm" (= millimeter) then the conversion to "m" (= meter) is totally acceptable.

All FMI parameters are defined in a configuration file (usually named as "**fmi_parameters.csv**"). Typically the SmartMet Server installation contains "default" FMI parameter definitions.

However, it is also easy to define own FMI parameters. This is usually done in a separate configuration file (for example "**ext/fmi_parameters.csv**"). In this case, the name of this file should be added into the main configuration file (usually named as "**grid-files.conf**").

2.3 FMI level type identifiers

A grid level type is needed in order to identify the type of the level value that is usually given in data queries. For example, it is typical that we have temperature information available in different level types. For example, we might have temperature information which level values are expressed as meters from the ground. On the other hand, we might have temperature values which level values are expressed as air pressure levels.

Different grid file formats use different identifiers of grid level types. That's why we have to map these format specific file level types to FMI level identifiers. All FMI level identifiers have unique names (= fmi-level-name) and numeric values (= fmi-level-id).

Here are some examples of the FMI level identifiers:

1	GROUND	Ground or water surface
2	PRESSURE	Pressure level
3	HYBRID	Hybrid level
4	ALTITUDE	Altitude
5	TOP	Top of atmosphere
6	HEIGHT	Height above ground in meters
7	MEANSEA	Mean sea level
8	ENTATM	Entire atmosphere
9	GROUND_DEPTH	Layer between two depths below land surface
10	DEPTH	Depth below some surface
11	PRESSURE_DELTA	Level at specified pressure difference from ground to level
12	MAXTHETA	Level where maximum equivalent potential temperature is found
13	HEIGHT_LAYER	Layer between two metric heights above ground
14	DEPTH_LAYER	Layer between two depths below land surface
15	ISOTHERMAL	Isothermal level, temperature in 1/100 K
16	MAXWIND	Maximum wind level

All FMI level identifiers are defined in a configuration file (usually named as "**fmi_levels.csv**"). Typically the SmartMet Server installation contains "default" FMI level definitions. However, it is also easy to define own FMI level. This is usually done in a separate configuration file (for example "**ext/fmi_levels.csv**"). In this case, the name of this file should be added into the main configuration file (usually named as "**grid-files.conf**").

2.4 FMI geometry identifiers

When parameters are queried over several different grid files, it is possible that exactly the same parameter from the same producer with the same time stamp and the same level can be found in multiple files. Usually the only difference is the actual grid geometry. For example, there might be a grid that contain data from the whole world, but the data resolution is very low. In addition, there might be a grid which geometrical area is smaller, but which data resolution is much higher.

If we want to make exact queries, we have to identify also different grid geometries. Technically we get this geometry information from grid files, which is actually a collection of different kinds of values that cannot be directly used in queries. That's why we need another mapping file that maps these geometries into numeric identifiers. This configuration file is usually named as "**fmi_geometries.csv**". The current file contains "default geometries" used by the SmartMet Server. If you want to define your own geometries, just create a new geometry file (for example, "**ext/fmi_geometries.csv**") and add its name into the main configuration file (usually named as "**grid-files.conf**").

3 GRIB Identifiers

3.1 Introduction

Different GRIB file versions (GRIB1, GRIB2, etc.) do not use same parameter or level identifiers. On the other hand, identification of parameters in GRIB files depends on several fields inside the file. Sometimes parameters can be identified with a combination of two fields, but sometimes the identification requires a combination of even ten fields.

In order to make any sense of this identification, we have to first map these fields into (global) GRIB parameter identifiers. This is an extra phase that we do not usually need to do with other grid file formats. After that we can map these GRIB parameter identifiers into the FMI parameter identifiers.

3.2 GRIB parameter identifiers

Luckily most of the (global) GRIB parameter identifiers are already defined in the ECMWF grib parameter database (<https://apps.ecmwf.int/codes/grib/param-db>). The SmartMet Server contains a partial copy of this information. GRIB parameter identifiers and definitions can be found in a configuration file (usually named as "**grib_parameters.csv**"). Notice that this file does not define actual mappings, it only defines GRIB parameters that are used in the mappings.

If necessary, it is possible to add own GRIB parameter identifiers. This is usually done only if there is no matching parameter definition in the "default" configuration file. In this case, new GRIB parameters should be added into a separate configuration file (for example "**ext/grib_parameters.csv**"). The name of this configuration file should be added into the main configuration file (usually named "**grid-files.conf**").

3.2.1 GRIB1 / GRIB2 field values => GRIB parameter identifiers

As was mentioned before, GRIB1 and GRIB2 parameters are identified with combinations of several (header) fields. We have to define how these fields can be mapped into (global) GRIB parameters.

Luckily most of these GRIB1/GRIB2 field mapping into GRIB parameter identifiers are already defined in the ECMWF grib parameter database (<https://apps.ecmwf.int/codes/grib/param-db>). The SmartMet Server installation contains a partial copy of this information. This means that the SmartMet Server can recognize the most of the widely used GRIB parameters in GRIB1 and GRIB2 files.

GRIB1 and GRIB2 use different fields for these mappings and that's why there are different mapping files for them (usually named "**grib1_parameters.csv**" and "**grib2_parameters.csv**"). If you want to create own mappings in addition to default mappings, just create your own mapping file and add its name into the main configuration file (usually named as "**grid-files.conf**").

GRIB1 mappings might look like this:

```
# Temperature
130;table2Version=200,indicatorOfParameter=11;
130;table2Version=1,indicatorOfParameter=11;
130;table2Version=160,indicatorOfParameter=130;

# U component of wind
131;table2Version=200,indicatorOfParameter=33;
131;table2Version=1,indicatorOfParameter=33;
131;table2Version=160,indicatorOfParameter=131;
```

The first field in the mapping file is the (global) GRIB parameter identifier. By using this identifier, we can find all information related to the current GRIB parameter (we defined this information earlier). The second field in the mapping file contains a list of (header) field values that are used for the mapping.

The same mappings with GRIB2 mappings might look like this:

```
# Temperature
130;discipline=0,parameterCategory=0,parameterNumber=0;

# U component of wind
131;discipline=0,parameterCategory=2,parameterNumber=2;
```

3.2.2 GRIB parameter identifiers => FMI parameter identifiers

At this point we should be familiar with the following things:

1. How FMI parameter identifiers are defined
2. How GRIB parameter identifiers are defined
3. How GRIB1 parameter identifiers are mapped into (global) GRIB parameter identifiers
4. How GRIB2 parameter identifiers are mapped into (global) GRIB parameter identifiers

The next thing to do is to map GRIB identifiers into FMI identifiers so that we can query information from GRIB files with FMI parameter names. These mappings are defined in a configuration file (usually named as "**fmi_parameterId_grib.csv**") and the configuration file looks like this:

```
# 130:Temperature (K)
T-K;130;;;
T-C;130;SUM{ $, 273.15 };SUM{ $, -273.15 };

# 131:U component of wind (m s-1)
U-MS;131;;;
```

The first field is the name (or id) of the FMI parameter and the second field is the GRIB parameter identifier. The third parameter is a conversion function required in order to convert the original GRIB parameter values into the FMI parameter values. The fourth field is the reverse conversion function that is required in order to convert FMI parameter values into the GRIB parameter values. No conversion is needed if these fields are empty.

If you want to define own "GRIB identifier => FMI identifier" mappings, you should use you own configuration file instead of the "default" configuration file. The idea is the same as with the other configuration files. Just create own configuration file into a different directory and add its name into the main configuration file (usually named as "**grid-files.conf**"). The point is that when you install a new version of the SmartMet Server, you will get new "default" configuration files, but your own additions are not overridden.

3.3 GRIB level type identifiers

In spite of that there is a numeric level type field available in GRIB1 and GRIB2 files, these fields do not use the same values. In addition, different producers might define the same level types differently than other producers, which means that we have to again map multiple (header) fields into a global level type identifier.

3.3.1 GRIB1 / GRIB2 field values => FMI level identifiers

In this time, we do not do any additional mappings in order to map these identifiers into (global) GRIB level identifiers, because there are no such (global) GRIB level information available (i.e. we have to define these mapping by ourselves). That's why we map these fields directly into FMI level identifiers.

GRIB1 and GRIB2 use different fields for these mappings and that's why there are different mapping files for them (usually named "**fmi_levelId_grib1.csv**" and "**fmi_levelId_grib2.csv**"). If you want to create own mappings in addition to default mappings, just create your own mapping file and add its name into the main configuration file (usually named as "**grid-files.conf**").

GRIB1 mappings might look like this:

```
1;indicatorOfTypeOfLevel=1;
2;indicatorOfTypeOfLevel=100;
3;indicatorOfTypeOfLevel=109;
4;indicatorOfTypeOfLevel=103;
5;indicatorOfTypeOfLevel=8;
6;indicatorOfTypeOfLevel=105;
6;generatingProcessIdentifier=1,centre=86,indicatorOfTypeOfLevel=1;
6;generatingProcessIdentifier=210,centre=86,indicatorOfTypeOfLevel=163;
7;indicatorOfTypeOfLevel=102;
7;generatingProcessIdentifier=2,centre=88,indicatorOfTypeOfLevel=103;
7;generatingProcessIdentifier=4,centre=88,indicatorOfTypeOfLevel=103;
8;indicatorOfTypeOfLevel=200;
```

The first field in the mapping file is the FMI level type identifier. By using this identifier, we can find all information related to the current FMI level type (we defined this information earlier). The second field in the mapping file contains a list of (header) field values that are used for the mapping.

GRIB2 mappings might look like this:

```
1;typeOfFirstFixedSurface=1;
2;typeOfFirstFixedSurface=100;
3;typeOfFirstFixedSurface=105;
5;typeOfFirstFixedSurface=8;
6;typeOfFirstFixedSurface=103;
7;typeOfFirstFixedSurface=101;
7;generatingProcessIdentifier=0,centre=251,typeOfFirstFixedSurface=102;
7;generatingProcessIdentifier=3,centre=251,typeOfFirstFixedSurface=102;
8;typeOfFirstFixedSurface=10;
8;generatingProcessIdentifier=96,centre=7,typeOfFirstFixedSurface=200;
9;typeOfFirstFixedSurface=106;
```

4 NetCDF Identifiers

4.1 Introduction

The current implementation of the NetCDF support is not complete, which means that only few projections are currently supported. On the other hand, only such NetCDF files are supported which header starts with 'CDF'. This means that we do not yet support NetCDF 4 files, which are actually in HDF5 format.

4.2 NetCDF parameter identifiers

The identification of the NetCDF parameters is based on the usage of "**standard_name**" attributes (<https://cfconventions.org/Data/cf-standard-names/current/build/cf-standard-name-table.html>). The value of this attribute is the same as the parameter identifier (netcdf-name). If a variable/parameter in the NetCDF file does not contain this attribute then it cannot be identified.

NetCDF parameter definitions can be found in a configuration file (usually named as "**netcdf_parameters.csv**"). This is just a list of standard NetCDF parameter names, units and descriptions. If necessary, it is possible to add own NetCDF parameter definitions. This is usually done only if there is no matching parameter definition in the "default" configuration file. In this case, new NetCDF parameters should be added into a separate configuration file and the name of this configuration file should be added into the main configuration file.

4.2.1 NetCDF parameter identifiers => FMI parameter identifiers

NetCDF parameter identifiers (= netcdf-name) are mapped into FMI parameter identifiers in the same way as the GRIB parameter identifiers are mapped into FMI parameter identifiers. These mappings are defined in a mapping file (usually named as "**fmi_parameterId_netcdf.csv**") and the file looks like this:

```
# P-PA => air_pressure (Pa)
P-PA;air_pressure;;;

# T-K => air_temperature (K)
T-K;air_temperature;;;
```

The first field is the name (or id) of the FMI parameter and the second field is the NetCDF parameter identifier. The third parameter is a conversion function required in order to convert the original NetCDF parameter values into the FMI parameter values. The fourth field is the reverse conversion function that is required in order to convert FMI parameter values into the NetCDF parameter values. No conversion is needed if these fields are empty.

If you want to define own "NetCDF parameter identifier => FMI parameter identifier" mappings, you should use your own configuration file instead of the "default" mapping file. Usually you should create own mappings 1) if the "default" mapping file does not contain the required mappings or 2) if you want to make mappings where you use a) own NetCDF parameters or b) own FMI parameters.

4.3 NetCDF level identifiers

Usually NetCDF files do not have clear level type definitions. Instead, they usually contain a list of levels and their units. At the moment we use these units in order to find a valid FMI level. So far, this is hardcoded, i.e. it cannot be configured. This might change in future when we continue the development of the NetCDF support.

5 Newbase Identifiers

5.1 Introduction

Finnish Meteorological Institute (=FMI) is using own very specific grid file format, which is called "querydata" or "qd". This file format was originally created, because FMI wanted to query grid information in a simple way. The main idea was that all other grid file formats (GRIB1, GRIB2, NetCDF, etc.) were converted to querydata format, and all queries were done over these querydata files. During the file conversion, all parameter and level identifiers were converted into so called "newbase identifiers". Also the actual grid data were converted to a new format. Sometimes also the grid resolution and projection were converted.

This arrangement had only one problem. The amount of data that Finnish Meteorological Institute is handling is increasing every day. This means that conversion of this data requires more and more time and processing power. Occasionally it might take hours to convert the latest data so that it can be queried. This is actually the reason why we now try to use original grid files directly without any conversion. In this approach we do not convert grid files, we just map their information into such form that we can execute queries over them. The point is that in this way we should be able to query information immediately when we get the latest grid files.

5.2 Newbase parameter identifiers

At the moment the SmartMet Server has different modules/engines taking care of different grids. In practice, so called "querydata engine" usually takes care of data queries over querydata files, meanwhile so called "grid engine" takes care of queries over GRIB1, GRIB2 and NetCDF files. The latest versions of the grid engine can also handle querydata files if this is wanted. All configurations and mappings that we have defined in this document are used by the grid engine. The querydata engine has different configuration files.

In spite of that the grid engine does not necessary use querydata files, there is a need to map Newbase parameter identifiers into FMI parameter identifiers. In this case, we want to make queries by using Newbase parameter names instead of FMI parameter names. The point is that Finnish Meteorological Institute has hundreds of scripts and tens of applications that are using Newbase parameter names in order to fetch data from querydata files. In future this data can be fetched from GRIB1, GRIB2 and NetCDF files by using the same Newbase parameter names, which means that there is no need to make changes into the existing scripts and the applications.

Newbase parameter identifiers are defined in a configuration file (usually named as "newbase_parameters.csv").

5.2.1 Newbase parameter identifiers => FMI parameter identifiers

As was mentioned earlier, we want to be able to use Newbase parameter identifiers for querying data that is usually queried with FMI parameter identifiers. That's why we should define mappings between these identifiers.

Newbase parameter identifiers (= netcdf-name) are mapped into FMI parameter identifiers in the same way as the GRIB parameter identifiers or NetCDF parameter identifiers are mapped into FMI parameter identifiers. These mappings are defined in a mapping file (usually named as "fmi_parameterId_newbase.csv") and the file looks like this:

```
# Temperature in Kelvins
T-K;Temperature;SUM{ $, -273.15 };SUM{ $, 273.15 };

# Temperature in Celsius
T-C;Temperature;;;
```

The first field is the name (or id) of the FMI parameter and the second field is the NetCDF parameter identifier, that can be Newbase parameter name or numeric id. The third parameter is a conversion function required in order to convert the original Newbase parameter values into the FMI parameter values. The fourth field is the reverse conversion function that is required in order to convert FMI parameter values into the Newbase parameter values. No conversion is needed if these fields are empty. In this example, it is good to realise that the Newbase parameter "Temperature" means temperature values in Celsius. That's why there are conversion functions in place when mapping between the FMI parameter "T-K" and the Newbase parameter "Temperature" is defined.

Technically these conversion functions are not needed when grid parameters are identified in the grid files. These definitions are needed when the grid engine queries parameters from the grid files. For example, if a user wants to query parameters by using Newbase identifiers then the query engine needs to find matching FMI identifiers for the requested Newbase identifiers. It is possible that there is no exact match between requested data and available data. For example, there might be a grid that contains temperature values in Kelvins, but if the user wants to get them in Celsius, then the grid engine needs to do the necessary conversion. Technically the grid engine uses different mapping files for converting FMI parameters to Newbase/NetCDF parameters. These mapping files are usually generated by using configuration and mapping files defined in this document.

If you want to define your own "Newbase parameter identifier => FMI parameter identifier" mappings, you should use your own configuration file instead of the "default" mapping file.

5.3 Newbase level type identifiers

Newbase level identifiers are mapped to FMI level identifiers by simple one-to-one mappings. These mappings are defined in a mapping file (usually named as "**fmi_levelId_newbase.csv**") and the file looks like this:

```
# 1;GROUND;Ground or water surface => kFmiGroundSurface (1)
1;1;

# 2;PRESSURE;Pressure level => kFmiPressureLevel (100)
2;100;

# 3;HYBRID;Hybrid level => kFmiHybridLevel (109)
3;109;
```

The first field is the FMI level identifier and the second field is the Newbase level identifier.

6 Summary

It is important to understand that all these configurations and mappings only define an information model that shows how different kinds of file format specific identifiers can be mapped into FMI identifiers. This information model itself does not do anything.

However, this information is usually needed when grid files are registered into the content database. The content database is the main information source when data is queried in the SmartMet Server. This means that the content of grid files needs to be registered into the content database before it can be queried. The content information is registered into the content database by using FMI identifiers.

Technically the SmartMet server does not care how these registrations are done. For example, the Finnish Meteorological Institute has an external database (called Radon) that contains information about all available grids. When we want to use these grids in the SmartMet Server, we copy the required registration information from this database into the content database. That's because this external Radon database is not a part of the SmartMet Server environment.

If there is no external information source available, we usually have to open grid files and pick the registration information directly from these files. The registration requires that we are able to map all file format specific identifiers into FMI identifiers. And that's why we have to define these mappings in the first place.