

Stinfosys

The metadata model of the Norwegian meteorological observations

by Gabriel Kielland and Terje Reite

Norwegian meteorological institute, P.O.Box 43 Blindern, N-0313 Oslo

Tel. +47 22 96 30 00, e-mail: gabriel.kielland@met.no

High quality metadata as well as high quality observations is essential in the assessment of the long time series necessary for climate variables.

Keeping track of sensors deployed and the varying observation methods enables the assessment of weaknesses, resolution and accuracy in the observations. The effort is on maintaining the physical station environment unchanged. Over monitoring periods of decades and centuries this is more of an ideal than reality. Stinfosys provides a framework for storing and retrieving the information as an ongoing process as it also supports the daily operational needs in its different aspects.

Metadata is registered as a distributed activity and shared across the institute on warehouse stock, sensor calibration and inspection reports, feedback on daily operations, correspondence, sensor specific parametrization in quality control, related station networks.

Stinfosys and metadata

The term metadata is used for two fundamentally different concepts¹. Structural metadata on the one hand is the design and specification of data structures. The data structures can be expressed as a domain model, conceptual schema or logical schema of an IT system. Descriptive metadata on the other hand, also referred to as metacontent, is about individual instances of application data, the data content. By the term "metadata model of meteorological observations" we here mean the structural metadata for an IT system used for storage of descriptive metadata of meteorological observations. In the following the term metadata is equal to the term descriptive metadata.

Stinfosys exists as a data system on its own terms and contains data that are registered as data. The content in Stinfosys can be viewed in a web interface or accessed via sql queries to the database, see Appendix 1 and figure 1.

What connects Stinfosys to the observations is the use of Stinfosys as metadata in relation to the observations. An example of that is the use of the Stinfosys data combined with a time series of observations. Another example where data from Stinfosys act as metadata is when Stinfosys is a tool to extract specific observation data series.

Observations

The total datamodel of the Norwegian meteorological observations is a table of observations plus the Stinfosys metadata model. The data table keys are stationid, paramid, hlevel, sensor, message_formatid and time. The column 'stationid' refers to a row in the table stinfosys.station that describes where the observation is done through the latitude and longitude and height above sea level at a certain time. The column 'paramid' refers to a row in the table stinfosys.param that describes the parameter monitored. The column 'hlevel' expresses a concept of formal height category, slightly different from the concept of physical height, in order to meet the needs for comparison and juxtaposition in quality controls and use of observations. The column 'sensor' is an abstract term for the sensor or equipment that actually did the measurement and allows parallel measurements. In some cases the same observations is transmitted in several ways, either from two different loggers or by different transmission techniques. The column 'message_formatid' refers to a row in the table stinfosys.message_format and allows parallel and simultaneous collection of observations.

The table stinfosys.sensor_info connects the observations further to other metadata and is the main observation definition table. sensor_info contains information about the equipment used to obtain the observations. The column 'sensor_info.equipmentid' connects the combination of stationid, paramid, hlevel and sensor in the observation data table to more information about the equipment. Stinfosys.sensor_info provides a reference to the measurement method and the detailed equipment model used; the instrument calibration status; instrument maintenance log; placement including physical height above ground or above reference height; ownership with details including whether the distribution of the observations are subject to restrictions.

Unifying internal needs

Every meteorological institution has necessarily some activity on weather station metadata. The ambition of the Norwegian project Stinfosys, initiated in 2002, was the integration of a multitude of lists and systems serving different but related purposes. To some extent the forecasters,

¹ See <http://en.wikipedia.org/wiki/Metadata>

climatologists, metrology staff, station logistics staff and programmers share elements of the same domain model. There are also differences, particularly in reference to aspects of time and the need for historic information. Information necessary in logistics of the present will prove essential for climate analyses decades later.

The purpose of Stinfosys is not only to have a metadata system on observations, but to have an information system on observations with data that are common between other systems and different uses. That is to have the information stored in a single repository, and subsequently export it to other places. Before Stinfosys, met.no had similar information stored in different databases. It was not exactly the same information and the domain worlds were different. That caused problems with the data flow of observations in our systems. So part of the development process was to clean up the inconsistencies. Part of the solution was to model the database more physicalistic and operationally with concepts that exists and in use outside the metadata framework as actual objects, measurement methods or administrative roles in the real world.

A consequence of this unifying method is the need to export metadata to other databases. There is an ongoing activity where the registration processes of other systems are gradually migrated into Stinfosys. In return jobs are set up to generate different metadata products since other systems not necessarily require the one-to-one content of the Stinfosys tables. Some systems require a mixture of modifications while other systems use a direct data dump. Examples of other systems that receive information from Stinfosys are Kvalobs for quality control of observations² and the climate databases where historical observations data series are stored.

In retrospect, the ten years specification and implementation period have been necessary to properly unify established routines and building trust in the unified approach.

The station

An important first step was a unification of station identification. Stinfosys inherited several older systems. The need for a common station identifier showed that even the basic concept 'weather station' is ambiguous. There is an inherent difference between the practical needs in logistics and a station concept based upon meteorological criteria. To what extent should instruments be separated in space and still be considered one station? To what extent should the station change without changing the identifier? To what extent can two institutions run instruments within the same station and with the same station identifier? A station is either fixed or mobile, but there are also hybrid stations in that respect like the weather ships of the past. On duty in position the weather ship would constitute a fixed station. Traveling to and from port the ship would be mobile, an important distinction with the weather observations metadata, fully unimportant in logistical operations around the ship.

The station name

Harmonization of the various naming practices was required. Inherently within the institute an agreement on name could be imposed, but there are a number of different user needs and requirements that needs to be considered. The data model should encompass any station run by other institutions. Thus a station name is not unique. The compromise solution of Stinfosys is to provide each station with an official name in upper case latin1 (ASCII plus the characters ÆØÅ) useful for database searches plus a second name for presentation purposes using the full Unicode character set and a limited number of characters. Separate from the two names the table network_station allows storage of any other station name variant with an explanation of its use. Examples are WMO registered names and airport names as registered with ICAO.

2 [http://www.wmo.int/pages/prog/www/IMOP/publications/IOM-82-TECO_2005/Papers/3\(13\)_Norway_Kielland.pdf](http://www.wmo.int/pages/prog/www/IMOP/publications/IOM-82-TECO_2005/Papers/3(13)_Norway_Kielland.pdf)
G. Kielland: Kvalobs. Presentation at WMO TECO, Bucuresti 7 May 2005

Aspects of time

In Stinfosys there are three concepts of time:

Edit time, the time of registration for each row in the database. At the end of each table in Stinfosys there are two columns on when this row was last updated or inserted and by whom respectively.

Most tables have the columns 'fromtime' and 'totime', used to express the time period the content of this column is valid. With the column 'fromtime' as part of the table key one has a possibility to deal with the history of that particular row. The column 'fromtime' cannot be NULL whereas when the column 'totime' is NULL it is considered currently valid and valid since the content of the column 'fromtime'. 'totime' is NULL also with planned activities. Not all the tables have this type of time. For example the table 'parameter' does not need this time as new additions never replace older variants. Also the table 'role' is kept without history indicating that a limited number of station network roles are registered consistently over time.

A third aspect of time is only for the observation programme, the table obspgm_h. Here each row expresses an observation expectation. The observation programme is expressed as a boolean array of length 24, one value true or false for each hour from 0 to 23, enabling the near real time detection of missing observations within the central receive system. There is also the possibility that the observation can come at any time at all.

Sensor information in observation control

Stinfosys provides information that can be processed and be part of the information used in data controls (kvalobs). In its most simple form the observation program in the table obspgm_h is used to detect missing observations. Less trivial is the use of data from the table sensor_info to control the observations. There are different sizes rain gauge buckets in the station network. An example is the use of threshold values on the bucket content in a range check for alarm purpose. Another example is the use of mounting height in the control of maximum snow depth measured by downward looking sensor.

The measurement method registered in the table sensor_info is also used to differentiate the parameter specific data controls according to measurement method.

From paper to files and database

The institute has produced and stored station metadata since its very beginning in 1866. Even with an established database and proper registration routines it would be an impossibly daunting task to have every historic metadata registered. The Norwegian approach is a hybrid solution with migration of information from paper documents into files and only eventually fully registered in the database. In parallel with the database development, old station documents, photos, drawings and maps were scanned. The scanned documents are stored in a separate file depository and referenced in the database by station, year and a content code. The set of content codes is divided between 35 different information aspects, see 2.

In the daily routine activity the redundant registration of database entries as well as files is useful in the assessment of metadata authenticity.

Reducing erroneous registration

The metadata production is inherently manual. The only automatization possible is whenever an external classification system is used such as national postal codes, European NUTs codes, and station lists maintained by other institutions like WMO. Otherwise every registration is prone to introduce random errors. This is conveniently improved by the synergy of the shared approach. All users are encouraged to report on errors detected. There are also scripts run to detect inconsistencies.

Technological aspects of the Stinfosys IT system

Stinfosys consists of both the technological system itself, its content, instructions on how to use the system and human routines for intended use of the system in an organizational context. The name of the technological system proper is here called the Stinfosys IT system. The Stinfosys IT system is based on Linux, web technology, a PostgreSQL database and files. The programming languages applied for the web development is both Perl and Java. Shell scripts are in limited use for cron jobs. Perl is the all round language for scripts and non-GUI programs. The Stinfosys GUIs are web forms for registration or for viewing content in the database and files.

In the foreseeable future we are interested in using both Perl and Java for web development of two different reasons:

- 1) met.no developers have different skills and different people master different programming languages.
- 2) the database and the data model should not be too strongly connected to one programming technology or GUIs. Over time both needs and technology will change, and changes to the data model are by far more costly than changes to the GUI.

Although PostgreSQL has the possibility to store files within the database, the files that are part of the Stinfosys system are stored separately. The reason is that a small database is easier to manage than a larger one, and pictures do take up a lot of space. There is currently some 90,000 files stored, half of which are pictures.

The code development is based on free software according to standard institute policy and the total cost considerations.

Appendix 1. Stinfosys data model schema

Appendix 2. Stinfosys codes on file semantic content

Figure 1. Stinfosys data model in UML format

Figure 2A. Operational weather station status as seen by the forecaster in the tool DIANA

Figure 2B. Example of Stinfosys metadata as displayed for the forecaster

Figure 2C. Example of Stinfosys station photo

Appendix 1

Stinfosys data model schema

```
TABLE stationid_list (  
  stationid INTEGER NOT NULL,  
  edited_by INTEGER NOT NULL,  
  edited_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY ( stationid )
```

);

DESCRIPTION of the table: This table contains the station numbers used.

DESCRIPTION of the table attributes:

stationid:

station numbers. The number 0 is reserved to be used for flags.

1) Norwegian national stations:

Norwegian national station number must be used only for Norwegian stations, station point or mobile station.

stationid range between 1 and 99999 (1 - 5 digits).

2) Foreign stations with WMO id.

It takes a WMO id adding two zeros at the end.

Foreign WMO stations:

WMO id = 2 digit WMO block number + 3 digit WMO station number.

Stationid range between 200100 and 9999900 (6-7 digits).

3) Non WMO stations:

For example, Swedish / Finnish precipitation stations, non-Norwegian ships, Norwegian stations without met.no national station numbers.

Stationid is a combination of:

- WMO block number 2 digits

- Organizationid 2 digits (for example: SMHI = 01)

- Local national station numbers - related to the "organization"

For Sweden, Finland and Iceland, the maximum number of digits in national station numbers are 3-5. National station number with four digits or less are increased to five digits by adding leading zeros.

Stationid range between 10100001 and 999999999 (8-9 digits).

4) Stations other than 1) or 2) or 3) have stationid sequentially selected from the values in the range between 1000000000 and 2147483647 (10 digits).

Non WMO stationid is autogenerated in Kvalobs with status static = "false" for random ships
edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of this row was last updated.

```
TABLE country (  
  countryid INTEGER NOT NULL,  
  name      TEXT NOT NULL,  
  alias     TEXT NOT NULL,  
  description TEXT DEFAULT NULL,  
  edited_by INTEGER NOT NULL,  
  edited_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY ( countryid )
```

);

DESCRIPTION of the table: Describes countries and territories by name, alias and eventually description. The table is not historical.

DESCRIPTION of the table attributes:

countryid: country code as used in WMOs Publication volume 9 A. Cf.

<http://www.wmo.int/pages/prog/www/ois/volume-a/9ALayoutGuide9805.html>

name: name of the country in the Norwegian language.

alias: a brief description of the name of the country as a memonics, ex. "NO", "SE" according to the standard ISO 3166-1 alpha-2.

description: a possible description of the country / territory.

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of this row was last updated.

TABLE municip (

municipid INTEGER NOT NULL,

code INTEGER NOT NULL,

name TEXT NOT NULL,

description TEXT DEFAULT NULL,

active BOOLEAN DEFAULT TRUE,

edited_by INTEGER NOT NULL,

edited_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,

PRIMARY KEY (municipid)

);

DESCRIPTION of the table: Describes the municipality or county with their name and any description of the municipality. The table is not historical, but contains disused ids.

DESCRIPTION of the table attributes:

municipid: four digits municipality number or two digits county number according to Norway statistics with met.no specific extensions to seas and overseas areas.

name: the name of the municipality or county.

description: a possible description of the municipality.

active: true when the municipid is in current use

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of this row was last updated.

TABLE postarea (

postcode INTEGER NOT NULL,

name TEXT NOT NULL,

municipid INTEGER NOT NULL,

edited_by INTEGER NOT NULL,

edited_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,

PRIMARY KEY (postcode)

);

DESCRIPTION of the table: This table gives the place name of a zip code The table is not historical.

DESCRIPTION of the table attributes:

postal code: ZIP code to the private address, as defined by the Postal Service.

name: city name that refers to the postcode.

municipid: See table municip.

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of this row was last updated.

TABLE protocol (

protocolid INTEGER NOT NULL,

name TEXT NOT NULL,

description TEXT DEFAULT NULL,

edited_by INTEGER NOT NULL,

edited_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,

PRIMARY KEY (protocolid)

);

DESCRIPTION of the table: list of communication protocols that are applied to persons, firms, drives and recorders (Landline, cell phone, e-mail, file transfer, TCP / IP)

DESCRIPTION of the table attributes:

protocolid: integer Protocol own code

name: the name of the protocol

description: description of the protocol

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of this row was last updated.

TABLE ontology (

ontologyid INTEGER NOT NULL,

description TEXT NOT NULL,

edited_by INTEGER NOT NULL,

edited_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,

PRIMARY KEY (ontologyid)

);

DESCRIPTION of the table: the metaphysical ontology of a station that expresses "degree of existence."

DESCRIPTION of the table attributes:

ontologyid: integer, own code.

description: describes the existence. A station can be real, complementary, synthetic or proxy.

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of this row was last updated.

TABLE station (

stationid INTEGER NOT NULL,

lat FLOAT DEFAULT NULL,

lon FLOAT DEFAULT NULL,

countryid INTEGER NOT NULL,

municipid INTEGER DEFAULT NULL,

Hs INTEGER DEFAULT NULL,

Hv INTEGER DEFAULT NULL,

Hp FLOAT DEFAULT NULL,

maxspeed FLOAT DEFAULT NULL,

name TEXT DEFAULT NULL,

short_name VARCHAR (16) DEFAULT NULL,

wmono INTEGER DEFAULT NULL,

ontologyid INTEGER NOT NULL,

fromtime TIMESTAMP NOT NULL,

totime TIMESTAMP DEFAULT NULL,

edited_by INTEGER NOT NULL,

edited_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,

PRIMARY KEY (stationid, fromtime),

FOREIGN KEY (stationid) REFERENCES stationid_list,

FOREIGN KEY (countryid) REFERENCES country,

FOREIGN KEY (municipid) REFERENCES municip,

FOREIGN KEY (ontologyid) REFERENCES ontology

);

DESCRIPTION of the table: A station is a point in space that has assigned different names and codes, where it will be associated with meteorological time series information. The station point can be fixed or mobile.

This table contains the description of the station point.

DESCRIPTION of the table attributes:

lat: geographic latitude, given in decimal degrees, WGS84 datum. Pursued accuracy is 1/100 arc

minutes.

lon: geographical longitude, given in decimal degrees, WGS84 datum. Pursued accuracy is 1/100 arc minutes.

countryid: country code, see table country.

municipid: municipality number, see table municip.

Hs: station point's height above mean sea level, given in meters.

Hs has one of four possible definitions:

- The altitude of the rain gauge's fixation point
- The altitude of the anemometer mast fixation point where there is no precipitation

measurement (Hs = Hv).

- The altitude of the screen fixation point where there is no precipitation measurement and no wind measurement

- Hs = Hp at airports and at complex instrument setup, but with an accuracy of meters.

All stations are required to have Hs registered.

Hv: the height of the anemometer mast fixation point, accuracy meters. Hv is relevant only where the wind mast is placed significantly far from the station point.

Hp: altitude of a fixed point characterized as the pressure sensor reference height, accuracy decimeter.

For airports Hp is the runway's highest point.

Other heights measured in meters above sea level, not stored in table station:

- HMI (the height of the fixing bolt MI) is at some stations used to calculate Hp. The number is stored as prose and referenced in table file_info.

- Hb is the altitude of the pressure sensor, accuracy decimeter, and is derived from hbe and Hp, where hbe = 'Hb - Hp'. hbe is stored as sensor_info.physical_height.

maxspeed: maximum speed of a vessel, measured in m/s. Immobile stations have maxspeed = 0.

name: Station name as nationally defined, with the Norwegian spelling limited to the latin1 character set, capitals. The only delimiter is a space and hyphen.

short_name: Short names such as related to requirements regarding distribution. May contain lower case letters and letters beyond latin1, as the Sami special characters.

wmono: Station Number including area number as registered with met.no, necessary for enrollment to WMOs systems (such as used in SYNOP messages) five digits. Wmono may be an official number or a preliminary assigned number not yet reported to the WMO.

The same official number as defined in WMOs systems can be found in the table network_station.

ontologyid: An id number that expresses the "degree of existence", see the table ontology.

fromtime: The time when the contents of the columns of the station began to apply; This means that it is not necessarily the time the station was created.

totime: The time when the contents of the columns of this station ceased to apply; This means that it is not necessarily the time the station was terminated.

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of this row was last updated.

```
TABLE docformat (  
  docformatid TEXT NOT NULL,  
  name TEXT NOT NULL,  
  description TEXT NOT NULL,  
  active BOOLEAN DEFAULT TRUE,  
  dismissreason TEXT DEFAULT NULL,  
  edited_by INTEGER NOT NULL,  
  edited_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY ( docformatid )  
);
```

DESCRIPTION of the table: describes the format used on a document

DESCRIPTION of the table attributes:

docformatid: id on format, own code system. docformatid = 0 means that the content resides in the field description and not in a separate file.

name: name of the format
description: description of the format such as limitations or version number.
active: specify whether the format is in active use, or whether it is historic.
dismissreason: Text for feedback to the operator about why the format is not approved for posting.
edited_by: personid of the person who entered or last modified the row.
edited_at: The time when the contents of this row was last updated.

```
TABLE stinfocode (
    code          TEXT NOT NULL,
    category      TEXT NOT NULL,
    description    TEXT DEFAULT NULL,
    short_description TEXT DEFAULT NULL,
    edited_by     INTEGER NOT NULL,
    edited_at     TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY ( code )
);
```

DESCRIPTION of the table: Table that defines sections for prose and text areas in the description of the station

DESCRIPTION of the table attributes:

code: a letter code that characterizes the contents of text and images in the description of the station, own code.

category: category relating to categories of the column 'code' in this table.

description: description and definition of the column 'code' in this table.

short_description: description of the column 'code' in this table adapted to HTML and alpha numeric sorting

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of this row was last updated.

```
TABLE active (
    activeid     INTEGER NOT NULL,
    description  TEXT NOT NULL,
    edited_by    INTEGER NOT NULL,
    edited_at    TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY ( activeid )
);
```

DESCRIPTION of the table: Help information about the validity of content for display

DESCRIPTION of the table attributes:

activeid: activeid = 1 indicates the the content is currently valid.

activeid = 0 indicates that the content is historically valid.

activeid = -1 indicates that the content is invalid and is still in the database for technical editorial reasons.

description: description of validity.

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of this row was last updated.

```
TABLE file_info (
    fileinfoid  SERIAL,
    stationid   INTEGER NOT NULL,
    year        INTEGER NOT NULL,
    code        TEXT NOT NULL,
    docformatid TEXT NOT NULL,
    description  TEXT DEFAULT NULL,
    filename    TEXT DEFAULT NULL,
);
```

```

minifilename TEXT DEFAULT NULL,
md5sum CHAR(32) NOT NULL,
activeid INTEGER DEFAULT 1,
edited_by INTEGER NOT NULL,
edited_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY ( fileinoid ),
FOREIGN KEY ( stationid ) REFERENCES stationid_list,
FOREIGN KEY ( code ) REFERENCES stinfocode,
FOREIGN KEY ( docformatid ) REFERENCES docformat,
UNIQUE (stationid, md5sum)

```

);

DESCRIPTION of the table: Table describing the station in prose as a reference to one document. The document is stored in a directory with the station number as directory name, and in a file with the combination of 'year' and 'code' as the first part of the filename. 'year' in its name refers to the year column in this table. The corresponding 'code' in its name refers to the column code in this table.

DESCRIPTION of the table attributes:

fileinoid: id of the document, serial number.

stationid: station number, see table stationid_list.

year: year in which this description was updated last time, four digits.

code: Station code documentation, see the table stinfocode.

docformatid: id for the file format, see the table docformat.

description: description of conditions relevant to the station, which is about the stationid and this year for this code. Used only for docformatid = 0.

filename: the exact filename of the file without a path.

minifilename: filename for the miniature file without a path.

md5sum: md5sum of the file in filename, or of description.

activeid: Code on the validity of the content, see the table active.

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of this row was last updated.

```

TABLE person (
  personid INTEGER NOT NULL,
  firstname TEXT NOT NULL,
  surname TEXT NOT NULL,
  username TEXT DEFAULT NULL,
  birthdate TIMESTAMP DEFAULT NULL,
  totime TIMESTAMP DEFAULT NULL,
  profession TEXT DEFAULT NULL,
  address TEXT DEFAULT NULL,
  postcode INTEGER DEFAULT NULL,
  edited_by INTEGER NOT NULL,
  edited_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY ( personid ),
  FOREIGN KEY ( postcode ) REFERENCES postarea

```

);

DESCRIPTION of the table: This table contains the standard information associated with a person.

DESCRIPTION of the table attributes:

personid: met.no employee number or other identification. For employees the number is in the range between 1 and 999999999 (1-9 digits). Other people have personid <0.

firstname: the person's name and any middle name.

surname: the person's last name.

username: user name to log on to computers, at met.no currently LDAP username.

birthdate: birth date.

totime: date the person's actual or potential tasks with the station network ceased.

profession: profession / education / skill description.
 address: the person's private mailbox or street address.
 postcode: ZIP code to the private address, see the table postarea.
 edited_by: personid of the person who entered or last modified the row.
 edited_at: The time from the contents of the row was last updated.

```
TABLE person_com (
  personid  INTEGER NOT NULL,
  protocolid INTEGER NOT NULL,
  eaddress  TEXT NOT NULL,
  description TEXT DEFAULT NULL,
  edited_by  INTEGER NOT NULL,
  edited_at  TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY ( personid, protocolid ),
  FOREIGN KEY ( personid ) REFERENCES person,
  FOREIGN KEY ( protocolid ) REFERENCES protocol
);
```

DESCRIPTION of the table: Describes the preferred means of communication to a person

DESCRIPTION of the table attributes:

personid: Id for person, see table person.

protocolid: id for the protocol to the electronic address, see table protocol.

eaddress: electronic address associated with the attribute "protocolid".

description: description of aspects of the communication method.

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of the row was last updated.

```
TABLE organisationid_list (
  organisationid INTEGER DEFAULT nextval("organisationid_list_nr_seq"::text) NOT NULL,
  edited_by      INTEGER NOT NULL,
  edited_at      TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY ( organisationid )
);
```

DESCRIPTION of the table: This table contains the organisationids in use.

DESCRIPTION of the table attributes:

organisationid: Id of the organization as integer, serial number.

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of the row was last updated.

```
TABLE organisation (
  organisationid INTEGER NOT NULL,
  countryid     INTEGER NOT NULL,
  name          TEXT NOT NULL,
  address       TEXT DEFAULT NULL,
  postcode      INTEGER DEFAULT NULL,
  fromtime      TIMESTAMP NOT NULL,
  totime        TIMESTAMP DEFAULT NULL,
  edited_by     INTEGER NOT NULL,
  edited_at     TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY ( organisationid, fromtime ),
  FOREIGN KEY ( organisationid ) REFERENCES organisationid_list,
  FOREIGN KEY ( countryid ) REFERENCES country,
  FOREIGN KEY ( postcode ) REFERENCES postarea
);
```

DESCRIPTION of the table: This table contains the description of an entity or group of people who met.no relate to in the context of stations

DESCRIPTION of the table attributes:

organisationid: Id of the organization as integer, serial number.

countryid: country code, see table country.

name: name of the entity or group.

address: the address of the entity, such as the mailbox or street.

postcode: postal code, see table postarea.

fromtime: The time when the contents of the row began to apply, which means that it is not necessarily the time the organization was created.

totime: The time when the contents of the row ceased to apply, which means that it is not necessarily the time the organization was disbanded.

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of the row was last updated.

```
TABLE organisation_com (  
    organisationid INTEGER NOT NULL,  
    protocolid    INTEGER NOT NULL,  
    eaddress     TEXT NOT NULL,  
    description  TEXT DEFAULT NULL,  
    edited_by    INTEGER NOT NULL,  
    edited_at    TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    PRIMARY KEY ( organisationid, eaddress ),  
    FOREIGN KEY ( organisationid ) REFERENCES organisationid_list,  
    FOREIGN KEY ( protocolid )    REFERENCES protocol
```

);

DESCRIPTION of the table: Describes the preferred means of communication for a business or other group of people

DESCRIPTION of the table attributes:

organisationid: Id of the organization as an integer, serial number.

protocolid: id for the protocol to the electronic address, see table protocol.

eaddress: electronic address associated with the attribute "protocolid".

description: description of aspects of the communication method.

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of this row was last updated.

```
TABLE role (  
    roleid INTEGER NOT NULL,  
    name   TEXT NOT NULL,  
    description TEXT NOT NULL,  
    edited_by    INTEGER NOT NULL,  
    edited_at    TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    PRIMARY KEY ( roleid )
```

);

DESCRIPTION of the table: This table defines the function of each individual roleid. One can look at this as roles or functions to each individual and/or entity in relation to a station; for example station keeper, observer, replacement, local technical assistant, salaried employee, landowner, inspector or in relation to equipment: for example instrument owner, borrower of an instrument, sender, receiver, forwarder.

or in relation to stinfosys I/O: for example user of an online report.

DESCRIPTION of the table attributes:

roleid: an integer that specifies an id for the role / function / responsibility.

name: the name of the role / function / responsibility.

description: a definition and description of the role / function / responsibility.

edited_by: personid of the person who entered or last modified the row.
edited_at: The time when the contents of this row was last updated.

```
TABLE contract (  
  contractid INTEGER DEFAULT nextval("contract_nr_seq"::text) NOT NULL,  
  name      TEXT DEFAULT NULL,  
  description TEXT DEFAULT NULL,  
  docformatid TEXT NOT NULL,  
  fileinfoid INTEGER DEFAULT NULL,  
  edited_by  INTEGER NOT NULL,  
  edited_at  TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY ( contractid ),  
  FOREIGN KEY ( docformatid ) REFERENCES docformat  
  FOREIGN KEY ( fileinfoid ) REFERENCES file_info  
);
```

DESCRIPTION of the table: Table of contracts / agreements for relations in the tables person_station, organisation_station, person_equipment, and organisation_equipment. This contract may be with the hiring / termination or other matters related to equipment. A contract is always associated with a role, but a role can have more than one contract.

The document (or collection of documents) are stored in a directory with the station number as a directory name, and in a file with four digit year and "Da" code as the first part of the filename, see the table file_info.

DESCRIPTION of the table attributes:

contractid: id for contract

name: name, title or keyword to the contract.

description: description of the contract. Contract text when docformatid = 0

docformatid: id for the format of the contract, see the table docformat.

fileinfoid: id of the document, serial number.

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of this row was last updated.

```
TABLE person_station (  
  personid  INTEGER NOT NULL,  
  stationid INTEGER NOT NULL,  
  roleid    INTEGER NOT NULL,  
  contractid INTEGER DEFAULT NULL,  
  description TEXT DEFAULT NULL,  
  fromtime  TIMESTAMP NOT NULL,  
  totime    TIMESTAMP DEFAULT NULL,  
  edited_by  INTEGER NOT NULL,  
  edited_at  TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY ( personid, stationid, roleid, fromtime ),  
  FOREIGN KEY ( personid ) REFERENCES person,  
  FOREIGN KEY ( stationid ) REFERENCES stationid_list,  
  FOREIGN KEY ( roleid ) REFERENCES role,  
  FOREIGN KEY ( contractid ) REFERENCES contract  
);
```

DESCRIPTION of the table: This table contains the relationship between a person and a station

DESCRIPTION of the table attributes:

personid: employee number or other identification, see the table person

stationid: station number, see table stationid_list.

roleid: id for the role, see the table role.

contractid: id for the contract / agreement, see the table contract.

description: description of the relationship to the extent necessary.

fromtime: The time when the person began to have this role in relation to the station.
 totime: The time when the person ceased to have this role to the station.
 edited_by: personid of the person who entered or last modified the row.
 edited_at: The time when the contents of this row was last updated.

```

TABLE organisation_station (
  organisationid INTEGER NOT NULL,
  stationid     INTEGER NOT NULL,
  roleid       INTEGER NOT NULL,
  contractid   INTEGER DEFAULT NULL,
  description   TEXT DEFAULT NULL,
  fromtime     TIMESTAMP NOT NULL,
  totime       TIMESTAMP DEFAULT NULL,
  edited_by    INTEGER NOT NULL,
  edited_at    TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY ( organisationid, stationid, roleid, fromtime ),
  FOREIGN KEY ( organisationid ) REFERENCES organisationid_list,
  FOREIGN KEY ( stationid )     REFERENCES stationid_list,
  FOREIGN KEY ( roleid )       REFERENCES role,
  FOREIGN KEY ( contractid )   REFERENCES contract
);

```

DESCRIPTION of the table: his table contains the relationship between an organization and a station

DESCRIPTION of the table attributes:

organisationid: Id of the organization, serial number.

stationID: station number, see table stationid_list.

roleid: id for the role, see the table role.

contractid: id for the contract / agreement, see the table contract.

description: description of the relationship to the extent necessary.

fromtime: The time when the organization began to have this role in relation to the station.

totime: The time when the organization ceased to have this role to the station.

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of this row was last updated.

```

TABLE equipmenttype (
  equipmenttypeid INTEGER NOT NULL,
  name            TEXT NOT NULL,
  description     TEXT DEFAULT NULL,
  issensor       BOOLEAN DEFAULT TRUE,
  hassoftware     BOOLEAN DEFAULT FALSE,
  edited_by      INTEGER NOT NULL,
  edited_at      TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY ( equipmenttypeid )
);

```

DESCRIPTION of the table: This table describes the type of equipment and what it used for

DESCRIPTION of the table attributes:

equipmenttypeid: id for the type of sensor / equipment.

name: name of the sensor category / equipment category.

description: Description of this category of equipment.

issensor: indicates whether the equipment can provide meteorological observations when installed in the station network.

If true (. true.) entry in the table sensor_info is relevant.

If false (. false.) entry in the table equipment_station is relevant.

hassoftware: indicate whether the equipment has software.

If true (. true.) entry in the table equipment_software is relevant.
 edited_by: personid of the person who entered or last modified the row.
 edited_at: The time when the contents of the row was last updated.

```
TABLE equipmentmodel (
  modelname          TEXT NOT NULL,
  equipmenttypeid   INTEGER NOT NULL,
  producerid        INTEGER NOT NULL,
  url                TEXT DEFAULT NULL,
  description        TEXT DEFAULT NULL,
  fromtime           TIMESTAMP DEFAULT NULL,
  totime            TIMESTAMP DEFAULT NULL,
  edited_by          INTEGER NOT NULL,
  edited_at          TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY ( modelname ),
  FOREIGN KEY ( equipmenttypeid ) REFERENCES equipmenttype,
  FOREIGN KEY ( producerid ) REFERENCES organisationid_list ( organisationid )
);
```

DESCRIPTION of the table: This table defines a list of equipment models. The definition of what constitutes a model is based on judgment as to which devices that are useful to stock.

DESCRIPTION of the table attributes:

model name: the name of the equipment model.

equipmenttypeid: id for the type of instrument / equipment, see table equipmenttype.

producerid: organisationid for the manufacturer, see the table organization.

url: URL to the description of the specific model.

description: Formal and pipe separated description of the model product.

A formal description is optional. The character '|' is used as a separator in the formal description.

The syntax is a key=value logic, where the area between two separators '|' is shared by using the delimiter '='. One might call this text area an attribute. What is to the left of the equals sign is the name of this attribute, or key. What is to the right is the content or value. It ends with delimiter '|'. After the last delimiter is any text. The order of attributes is irrelevant, except for the text that must be at the end. There is only one attribute when the delimiter '|' is missing, either a text or a key/value pair. The name of the attributes (i.e. key) is in English and is one word only for the purpose of future formalization and parsing. The keys are reserved words that cannot be used in free text.

The following keys are defined:

resolution

unit_resolution

correction_formula

correction_constant_A = model specific constant for tilt pluviometers for the conversion of tilts per minute to mm precipitation per minute.

correction_constant_B = model specific constant for tilt pluviometers for the conversion of tilt per minute to mm precipitation per minute.

These constants are included in the formula:

mm precipitation per minute = correction_constant_A*tilts + correction_constant_B*tilts*tilts

fromtime: time this model was produced for the first time or first time it was used on the drive.

totime: the time of this model is no longer available for deployment at the station.

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of the row was last updated.

```
TABLE equipment (
  equipmentid        INTEGER DEFAULT nextval("equipment_nr_seq"::text) NOT NULL,
  modelname          TEXT NOT NULL,
  serialno           TEXT DEFAULT NULL,
  edited_by          INTEGER NOT NULL,
```



```
    edited_at    TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    PRIMARY KEY ( equipmentid ),  
    FOREIGN KEY ( modelname ) REFERENCES equipmentmodel
```

);

DESCRIPTION of the table: This table includes all equipment.

DESCRIPTION of the table attributes:

equipmentid: id for the equipment, serial number.

modelname: name of the equipment, see table equipmentmodel.

serialno: serial number.

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of the row was last updated.

```
TABLE equipmentstatusdef (
```

```
    statusid    INTEGER NOT NULL,  
    name        TEXT NOT NULL,  
    ex_name     TEXT DEFAULT NULL,  
    description TEXT NOT NULL,  
    edited_by   INTEGER NOT NULL,  
    edited_at   TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    PRIMARY KEY ( statusid )
```

);

DESCRIPTION of the table: This table defines the conditions of a device's life cycle.

The classification is appropriate in relation to the station network department procedures.

DESCRIPTION of the table attributes:

statusid: id code that defines the status of the equipment.

name: the name of the state expressed in the present tense.

ex_name: name of the state expressed in the past tense.

description: description and definition of the status.

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of the row was last updated.

```
TABLE equipment_status (
```

```
    equipmentid INTEGER NOT NULL,  
    statusid    INTEGER NOT NULL,  
    totime      TIMESTAMP DEFAULT NULL,  
    fromtime    TIMESTAMP NOT NULL,  
    edited_by   INTEGER NOT NULL,  
    edited_at   TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    PRIMARY KEY ( equipmentid, fromtime ),  
    FOREIGN KEY ( equipmentid ) REFERENCES equipment,  
    FOREIGN KEY ( statusid ) REFERENCES equipmentstatusdef
```

);

DESCRIPTION of the table: This table includes all equipment related to an event.

DESCRIPTION of the table attributes:

equipmentid: id for the equipment, serial number.

statusid: id of device status, see the table equipmentstatusdef.

totime: The time (date) when the equipment ceased to have the status indicated by statusid

fromtime: The time (date) when the equipment started to have the status indicated by statusid.

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of the row was last updated.

```
TABLE person_equipment (
```

```
    personid    INTEGER NOT NULL,
```

```

equipmentid  INTEGER NOT NULL,
roleid       INTEGER NOT NULL,
contractid   INTEGER DEFAULT NULL,
description  TEXT DEFAULT NULL,
edited_by    INTEGER NOT NULL,
edited_at    TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY ( personid, equipmentid, roleid ),
FOREIGN KEY ( personid ) REFERENCES person,
FOREIGN KEY ( equipmentid ) REFERENCES equipment,
FOREIGN KEY ( roleid ) REFERENCES role,
FOREIGN KEY ( contractid ) REFERENCES contract

```

);

DESCRIPTION of the table: This table contains the relationship between equipment and person.

DESCRIPTION of the table attributes:

personid: employee number or other identification, see the table person.

equipmentid: id to locate the instrument in the tables that have the equipment, see table equipment.

roleid: id for the role, for example instrument owner, borrower of instrument, sender, receiver, forwarder, see the table role.

contractid: id for the contract, see the table contract.

description: description of the relationship to the extent necessary.

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of the row was last updated.

```

TABLE organisation_equipment (
  organisationid INTEGER NOT NULL,
  equipmentid    INTEGER NOT NULL,
  roleid         INTEGER NOT NULL,
  contractid     INTEGER DEFAULT NULL,
  description    TEXT DEFAULT NULL,
  edited_by     INTEGER NOT NULL,
  edited_at      TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY ( organisationid, equipmentid, roleid ),
  FOREIGN KEY ( organisationid ) REFERENCES organisationid_list,
  FOREIGN KEY ( equipmentid ) REFERENCES equipment,
  FOREIGN KEY ( roleid ) REFERENCES role,
  FOREIGN KEY ( contractid ) REFERENCES contract

```

);

DESCRIPTION of the table: This table contains the relationship between the equipment and organization

DESCRIPTION of the table attributes:

organisationid: Id of the organization, see the table organisationid_list.

equipmentid: id to locate the instrument in the tables that have the equipment, see table equipment.

roleid: id for the role, for example instrument owner, receiver, forwarder, see the table role.

contractid: id for the contract, see the table contract.

description: description of the relationship to the extent necessary.

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of the row was last updated.

```

TABLE param (
  paramid      INTEGER NOT NULL,
  name         TEXT DEFAULT NULL,
  description  TEXT DEFAULT NULL,
  weather_element TEXT DEFAULT NULL,
  unit        TEXT DEFAULT NULL,

```

```

hlevel_scale  INTEGER DEFAULT 0,
standard_hlevel INTEGER DEFAULT NULL,
standard_physical_height TEXT DEFAULT NULL,
comment      TEXT DEFAULT NULL,
scalar       BOOLEAN DEFAULT TRUE,
edited_by    INTEGER NOT NULL,
edited_at    TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY ( paramid )

```

);

DESCRIPTION of the table: Code and description of all entities physically parameterized by the Norwegian Meteorological Institute and the relevant to meteorology. These parameters include both qualitative human observations of weather abbreviated into codes, physical quantities actually measured, and derived physical quantities.

DESCRIPTION of the table attributes:

paramid: parameter id. The value 0 is reserved and shall not be a separate parameter.

name: parameter alphanumeric code

description: Parameter Description

weather_element: says something about what kind of weather element or related physical entity this is such as temperature, wind, pressure; pressure tendency is the same as weather element

pressure, but a different physical quantity

unit: The unit of parameter

hlevel_scale: This is a tier-exponent scaling factor for this hlevel parameter, to interpret hlevel in the tables sensor_info and obspgm_h, and table data in kvalobs.

Meters have a value of 0, for cm -2, for km, we will have 3, etc.

standard_hlevel: This attribute contains values for the default value hlevel for the parameter where hlevel = 0

This is interpreted as the height from the ground with the exception of wind that is interpreted as the height above the highest terrain feature.

standard_physical_height: Values and textual descriptions of what is the standard height for physical parameter which hlevel = 0. This will often involve a range of valid values.

comment: Additional text on specific aspects of the parameter.

scalar: indicates whether the parameter values are numbers.

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of the row was last updated.

TABLE measurement_method (

```

measurement_methodid INTEGER NOT NULL,
paramid              INTEGER NOT NULL,
description          TEXT DEFAULT NULL,
edited_by           INTEGER NOT NULL,
edited_at           TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY ( measurement_methodid, paramid ),
FOREIGN KEY ( paramid ) REFERENCES param

```

);

DESCRIPTION of the table: This table shows the measurement method and the various methods applied to measure the same parameter

DESCRIPTION of the table attributes:

measurement_methodid: Id showing the measurement method for a particular parameter, including native code based on BUFR Table 0 02 175 a.o.

paramid: paramid that this method of measurement applies to, see table param

description: Description of the measuring method in prose.

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of the row was last updated.

```

TABLE equipment_station (
    equipmentid    INTEGER NOT NULL,
    stationid      INTEGER NOT NULL,
    place          TEXT DEFAULT NULL,
    description    TEXT DEFAULT NULL,
    totime        TIMESTAMP DEFAULT NULL,
    fromtime      TIMESTAMP NOT NULL,
    edited_by     INTEGER NOT NULL,
    edited_at     TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY ( equipmentid, stationid, fromtime ),
    FOREIGN KEY ( equipmentid ) REFERENCES equipment,
    FOREIGN KEY ( stationid ) REFERENCES stationid_list
);

```

DESCRIPTION of the table: the link between the equipment and station where the equipment do not perform measurements. Similarly, coupling between the measuring equipment and station are in the table sensor_info.

DESCRIPTION of the table attributes:

equipmentid: id to the equipment, see table equipment.

stationid: station number, see table stationid_list.

place: textual description of where the station equipment is located.

description: description of equipment on the station.

totime: date equipment at the station was removed from the station.

fromtime: date the equipment was mounted at the station or a change in the attribute "place" or a change in the attribute "description".

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the row was last updated.

```

TABLE sensor_info (
    equipmentid    INTEGER NOT NULL,
    paramid       INTEGER NOT NULL,
    stationid     INTEGER NOT NULL,
    sensor        INTEGER DEFAULT NULL,
    hlevel        INTEGER DEFAULT NULL,
    operational    BOOLEAN NOT NULL,
    physical_height FLOAT DEFAULT NULL,
    place         TEXT DEFAULT NULL,
    description    TEXT DEFAULT NULL,
    measurement_methodid INTEGER DEFAULT NULL,
    distributable TEXT DEFAULT NULL,
    fromtime      TIMESTAMP NOT NULL,
    totime        TIMESTAMP DEFAULT NULL,
    edited_by     INTEGER NOT NULL,
    edited_at     TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY ( equipmentid, paramid, fromtime ),
    FOREIGN KEY ( equipmentid ) REFERENCES equipment,
    FOREIGN KEY ( paramid ) REFERENCES param,
    FOREIGN KEY ( stationid ) REFERENCES stationid_list,
    FOREIGN KEY ( measurement_methodid, paramid ) REFERENCES measurement_method
);

```

DESCRIPTION of the table: A table of sensors that are deployed and / or installed in the station network. The table brings together concepts of measurements and instruments.

DESCRIPTION of the table attributes:

equipmentid: id for equipment, see table equipment.

paramid: parameter numbers, see table param. A limited range of parameters are used, corresponding to the subdivision by param.weather_element.

stationid: station number, see table stationid_list.

sensor:

relative id for the sensor. stationid, paramid, hlevel and sensor defines the total sensor concept. Sensor number where the same parameter for the same height (and the same station) is measured from several sensors. Sensor = 0 shall indicate the sensor which is the highest priority for use.

hlevel:

height above the ground for the sensor if it is not measured in standard height. hlevel=0 indicates measurement in standard height, or that height measurement is irrelevant, which is perceived as a symbolic value.

hlevel = 0 is default for data collection.

Standard heights is defined in the table param.

The actual physical height is defined in the attribute "physical_height" in this table. hlevel means that the physical height in "physical_height" belong to the same category and can be compared.

operational: boolean variable that says if the sensor is in use as a measuring instrument or not.

physical_height: actually measured physical height of the instrument above the ground or above relevant reference level Hp, Hv.

place: textual description of where the equipment is located within the station.

description: Formal and pipe separated description of this table.

A formal description is optional. The character '|' is used as a separator in the formal description.

The syntax is a key=value logic, where the area between two separators '|' is shared by using the delimiter '='. One might call this text area an attribute. What is to the left of the equals sign is the name of this attribute, or key. What is to the right is the content or value. It ends with delimiter '|'.

After the last delimiter is any text. The order of attributes is irrelevant, except for the text that must be at the end. There is only one attribute when the delimiter '|' is missing, either a text or a key/value pair. The name of the attributes (i.e. key) is in English and is one word only for the purpose of future formalization and parsing. The keys are reserved words that cannot be used in free text.

Typical content of the text:

Description of equipment usage in the station.

Description of the sensor, changes made;

measurement_methodid: measurement method, see the table measurement_method.

distributable: description of any limitations on distributing the measurement values outside met.no core service.

fromtime: time the sensor starts sending or time of last modification of the contents of the row (especially for the column 'physical_height' with same hlevel, equipmentid, description, measurement_methodid, operational).

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of the row was last updated.

```
TABLE equipment_config (  
    equipmentid  INTEGER NOT NULL,  
    eaddress     TEXT DEFAULT NULL,  
    protocolid   INTEGER NOT NULL,  
    description  TEXT NOT NULL,  
    fromtime     TIMESTAMP NOT NULL,  
    edited_by    INTEGER NOT NULL,  
    edited_at    TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    PRIMARY KEY ( equipmentid, protocolid, fromtime ),  
    FOREIGN KEY ( equipmentid ) REFERENCES equipment,  
    FOREIGN KEY ( protocolid ) REFERENCES protocol  
);
```

DESCRIPTION of the table: Table describing the configuration of equipment

DESCRIPTION of the table attributes:

equipmentid: id for equipment, see table equipment.

eaddress: electronic address associated with the protocol represented by protocolid, for example phone number, mobile number, e-mail address).

protocolid: id for different protocols, see the table protocol, for example telephone, cell phone, e-mail).

description: The configuration in ASCII format, eg. PIN(s) of a cell phone.

from time: the date of this configuration starts / started to apply.

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of the row was last updated.

```
TABLE message_in (  
  stationid INTEGER NOT NULL,  
  message_formatid INTEGER NOT NULL,  
  kvalobsid INTEGER NOT NULL,  
  fromtime  TIMESTAMP NOT NULL,  
  totime    TIMESTAMP DEFAULT NULL,  
  edited_by INTEGER NOT NULL,  
  edited_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY ( stationid, message_formatid, kvalobsid, fromtime ),  
  FOREIGN KEY ( stationid ) REFERENCES stationid_list,  
  FOREIGN KEY ( kvalobsid ) REFERENCES kvalobs
```

);

DESCRIPTION of the table: Table that provides control over the routing of messages from the receipt in the central system into different instances of the control system kvalobs

DESCRIPTION of the table attributes:

stationid: station number, see table stationid_list.

stationid = 0 means that all messages in this format is to be routed to the same kvalobs instance.

message_formatid: Id of the message format, see the table message_format.

kvalobsid: Id of the instance of kvalobs, see the table kvalobs.

fromtime: The time when the contents of the row begins / began to apply.

to time: The time the contents of this row is no longer valid.

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of the row was last updated.

```
TABLE kvalobs (  
  kvalobsid INTEGER NOT NULL,  
  name TEXT NOT NULL,  
  description TEXT DEFAULT NULL,  
  edited_by INTEGER NOT NULL,  
  edited_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY ( kvalobsid )
```

);

DESCRIPTION of the table: The table explains the various instances of the quality system kvalobs referenced in the table message_in

DESCRIPTION of the table attributes:

kvalobsid: Id of the instance of kvalobs

name: The name of the instance of kvalobs

description: Description of the relevant factors in relation to the instance of kvalobs

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of the row was last updated.

```
TABLE station_outmessage (  
  stationid      INTEGER NOT NULL,  
  productcoupling TEXT DEFAULT NULL,  --AUTO: typepriority  
  coupling_delay TEXT DEFAULT NULL,  --AUTO (kvsynopd): delay  
  priority_precip TEXT DEFAULT NULL,  --AUTO: precipitation
```

```

fromtime      TIMESTAMP NOT NULL,
totime        TIMESTAMP DEFAULT NULL,
edited_by     INTEGER NOT NULL,
edited_at     TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY ( stationid, fromtime ),
FOREIGN KEY ( stationid ) REFERENCES stationid_list

```

);

DESCRIPTION of the table: A table provides guidelines for the production of the official real time message of met.no observations.

DESCRIPTION of the table attributes:

stationid: station number, see table stationid_list.

productcoupling: With the generation of product it is a order of priority for which format should be selected for a given parameter. The format chosen is that which has highest priority for that particular parameter in the incoming data.

The field syntax is a comma-separated list of message_formatid. (See table message_format). An asterisk in front indicates that the message format marked is required for the product to be generated.

coupling_delay: The maximum time the linked product may delay before all part have arrived. Time is measured in minutes

priority_precip: If there is more than one precipitation parameter this attribute assigns which of them to be used by priority in the official real-time product.

The codes are: "RA" – precipitation calculated from change in a bucket value, "RRRtR" - initially reported parameter tR, "RR_1" – hourly precipitation amount is calculated in the station logger.

fromtime: The time when the contents of the row begins / began to apply.

totime: The time the contents of this row is no longer valid.

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of the row was last updated.

```

TABLE message_format (
  message_formatid INTEGER NOT NULL,
  name              TEXT DEFAULT NULL,
  earlyobs          INTEGER DEFAULT NULL,
  lateobs           INTEGER DEFAULT NULL,
  read              TEXT DEFAULT NULL,
  frequency         TEXT DEFAULT NULL,
  description       TEXT DEFAULT NULL,
  edited_by         INTEGER NOT NULL,
  edited_at         TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY ( message_formatid )

```

);

DESCRIPTION of the table:

The format characterizing the communication between the station and met.no. This means that the logger at the station sends a message that is parsed at met.no. The format of this message is the message format.

DESCRIPTION of the table attributes:

message_formatid: A number that identifies the typical features of the data from a weather station that is due to different readings, format used for transmission / dispatch of the data and registration of the reading information from the sensor.

name: The name of the message format.

earlyobs: time tolerance in minutes for early arrival of the message.

lateobs: time tolerance in minutes for late arrival of the message.

read: Code for reading method. The method is the default in use with the application of the message format. Types: M = Manual A = Automatic, I = (Kvalobs) Internal. Reading method is not the same as the sending method.

frequency: typical interval in the observation program, the hour or minute or any half hour.

description: comment or description of the format.
 edited_by: personid of the person who entered or last modified the row.
 edited_at: The time when the contents of the row was last updated.

```
TABLE obspgm_h (
  stationid      INTEGER NOT NULL,
  paramid        INTEGER NOT NULL,
  hlevel         INTEGER NOT NULL,
  message_formatid INTEGER NOT NULL,
  nsensor        INTEGER DEFAULT 1,
  priority_message BOOLEAN DEFAULT TRUE,
  anytime        BOOLEAN DEFAULT FALSE,
  hour           BOOLEAN[24] DEFAULT '{FALSE, FALSE, FALSE, FALSE, FALSE, FALSE,
FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE,
FALSE, FALSE, FALSE, FALSE, FALSE, FALSE}',
  test           BOOLEAN DEFAULT FALSE,
  totime         TIMESTAMP DEFAULT NULL,
  fromtime       TIMESTAMP NOT NULL,
  edited_by      INTEGER NOT NULL,
  edited_at      TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY ( stationid, paramid, hlevel, message_formatid, fromtime ),
  FOREIGN KEY ( stationid ) REFERENCES stationid_list,
  FOREIGN KEY ( paramid ) REFERENCES param,
  FOREIGN KEY ( message_formatid ) REFERENCES message_format
);
```

DESCRIPTION of the table: This table is the observation program (i.e. Kvalobs' control program) for a station, based on hourly messages or less. The observation program describes when a parameter observed in a station is expected. Not all parameters are observed for all times. Also a station (observer) do not need to observe at all times.

This table indicates when a parameter is observed. The content is used to determine if an observation has not been received.

DESCRIPTION of the table attributes:

stationid: station number, see the table stationid_list.

paramid: parameter number, see the table param.

hlevel: the relative height of the sensor, see the table sensor_info.

message_formatid: Id of the message format, see the table message_format.

nsensor: number of sensors for a given combination of paramid, stationid and hlevel for a given fromtime. Example: nsensor has the value 3 as it has 3 sensors that have the value 0, 1, 2.

What these numbers mean is defined in the table sensor_info.

priority_message: indicates whether the message format is the priority for a given combination of stationid, paramid and hlevel. priority_message='true' means that the observation with given message_formatid should be used for distribution.

anytime: The value anytime='true' indicates that the observation may enter at any time.

The value anytime='false' specifies that the column 'hour' is relevant and describing the hours when the observation is scheduled.

Test: true if the given combination of stationid, paramid, hlevel and message_formatid is a test observation, false otherwise.

hour: boolean [24] observation schedule, on the hour.

totime: The time the contents of this row is no longer valid.

fromtime: The time when the data for that row begin / began to apply.

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of the row was last updated.

```
TABLE obs_network (
```



```

networkid      INTEGER NOT NULL,
name           TEXT DEFAULT NULL,
codename       TEXT DEFAULT NULL,
description    TEXT DEFAULT NULL,
edited_by      INTEGER NOT NULL,
edited_at      TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY ( networkid )

```

);

DESCRIPTION of the table: Table of one or more networks of selected stations. Networks can be defined both internationally and nationally. The table does not include network history.

DESCRIPTION of the table attributes:

networkid: id for the network.

name: the name of the network.

codename: The network station code designation.

description: description of the network.

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of the row was last updated.

```

TABLE network_station (
  stationid      INTEGER NOT NULL,
  networkid      INTEGER NOT NULL,
  name           TEXT DEFAULT NULL,
  external_stationcode TEXT DEFAULT NULL,
  comment        TEXT DEFAULT NULL,
  totime         TIMESTAMP DEFAULT NULL,
  fromtime       TIMESTAMP NOT NULL,
  edited_by      INTEGER NOT NULL,
  edited_at      TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY ( stationid, networkid, fromtime ),
  FOREIGN KEY ( stationid ) REFERENCES stationid_list,
  FOREIGN KEY ( networkid ) REFERENCES obs_network

```

);

DESCRIPTION of the table: Table of the stations included in one or more network

DESCRIPTION of the table attributes:

stationid: station number, see table stationid_list.

networkid: id for the network, see the table obs_network.

name: the station name within the specific network, often the same as station.name

external_stationcode: id for the station associated with the remote station network or another number system. Cf. obs_network.codename.

comment: any comments associated with the station's network affiliation

totime: The time when the station's network affiliation ends / closed

fromtime: The time when the station's network affiliation begins / began

edited_by: personid of the person who entered or last modified the row.

edited_at: The time when the contents of the row was last updated.

Appendix 2

Stinfosys codes on file semantic content

The documents are stored in file folders with national station number for the folder name. The file name is a combination of four digit year, followed by one to two letters content code, followed by a text to further explain the document content. The year indicates when the document's content was updated last time. The table shows the codes for document content in order of priority. If a document contains several types of content, only one code is used selected from the top of the list downwards.

Type of document	Code	Explanation
Photo	Fh	Photo of horizon panorama
	Fi	Photo of station sensor placement in the terrain
	Fd	Photo of detail / close-up (such as sensor)
	Fk	Photo connectors / cabling / termination
	Fo	Photo of station surroundings
	Fs	Other station photo (such as people)
Map	Kt	Topography map (scale 1:2000 and smaller)
	Ko	Map of site details (scale range 1:200 - 1:2000)
	Kf	Map with fixed points (for visual range)
Drawing, also computer assisted	Ho	Drawing of station surroundings
	Hf	Drawing of fixed points (for visual range)
	Hs	Drawing of station sensor placement
	Hi	Drawing of instruments
	Hk	Drawing of connectors / cabling / termination
	Hh	Drawing of horizon panorama
Text	Bs	Assessment of station setup / representativity
	Bo	Area description (terrain surrounding)
	Be	Assessment of the instruments' exposure
	Tk	Connectors / cabling / termination as text
	Ti	Instrument inventory (for example with serial number)
	TI	Logger configuration
	Bp	Description of people (contact information for the observer or local contact)
	Bk	Communications, access, data communications
	Bh	History
	Th	Description of facilities (other than sensors)
	Tb	Other technical description (including fixed points for levelling, description of the station's altitude)
	I	Inspection report in general (such as instrument control, amendments, instructions, preliminary inspection report, test results)
	Q	control / comment on the inspection report (other than inspector)
	Document	Da
Dr		Report (including press cuttings)
Dc		Instructions / guidelines (to the observer)
Db		Letter including bills, invoices (for companies outside the Norwegian Meteorological Institute)
Dm		Message (between Institute staff and with the observer)
Dd		Miscellaneous

Figure 2A. Operational weather station status as seen by the forecaster in the tool DIANA

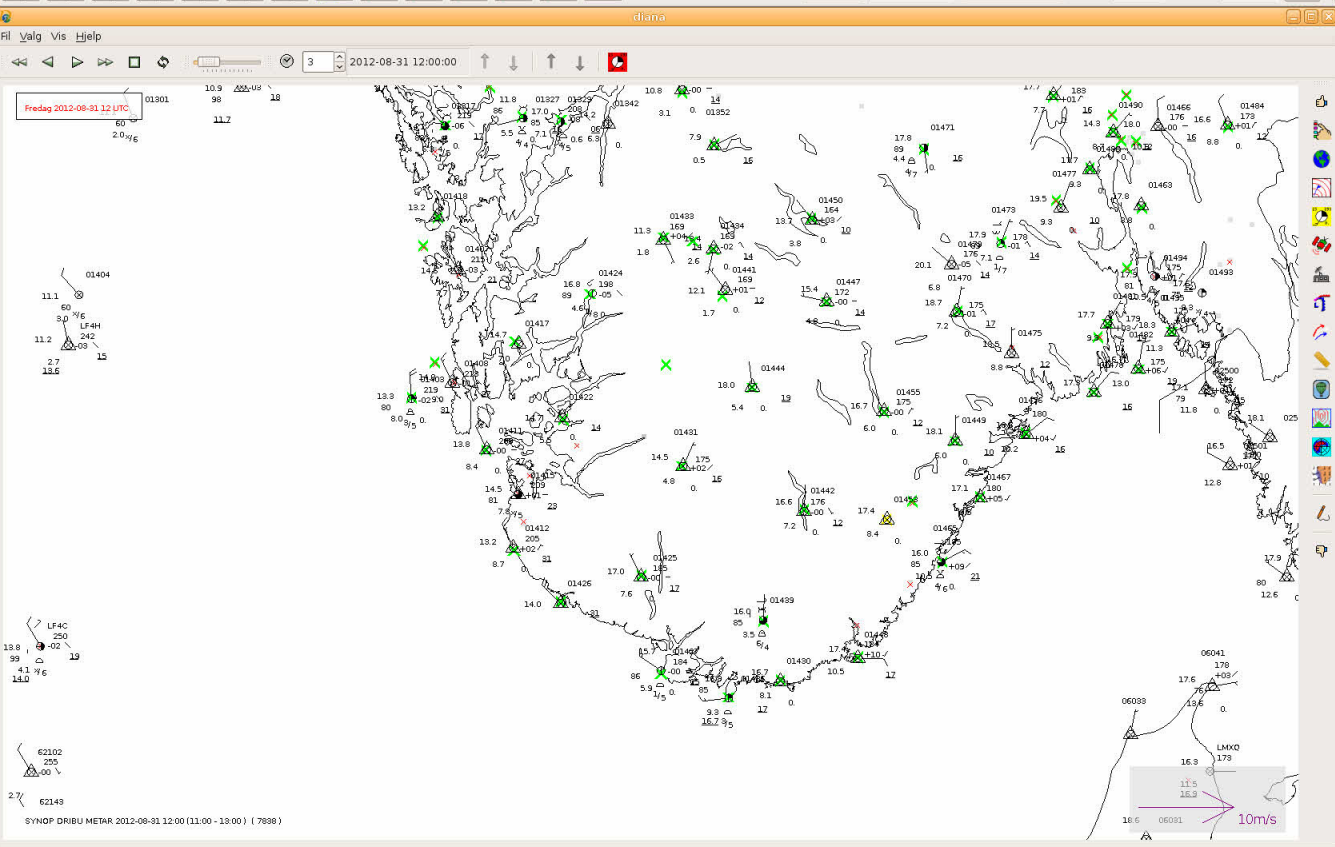
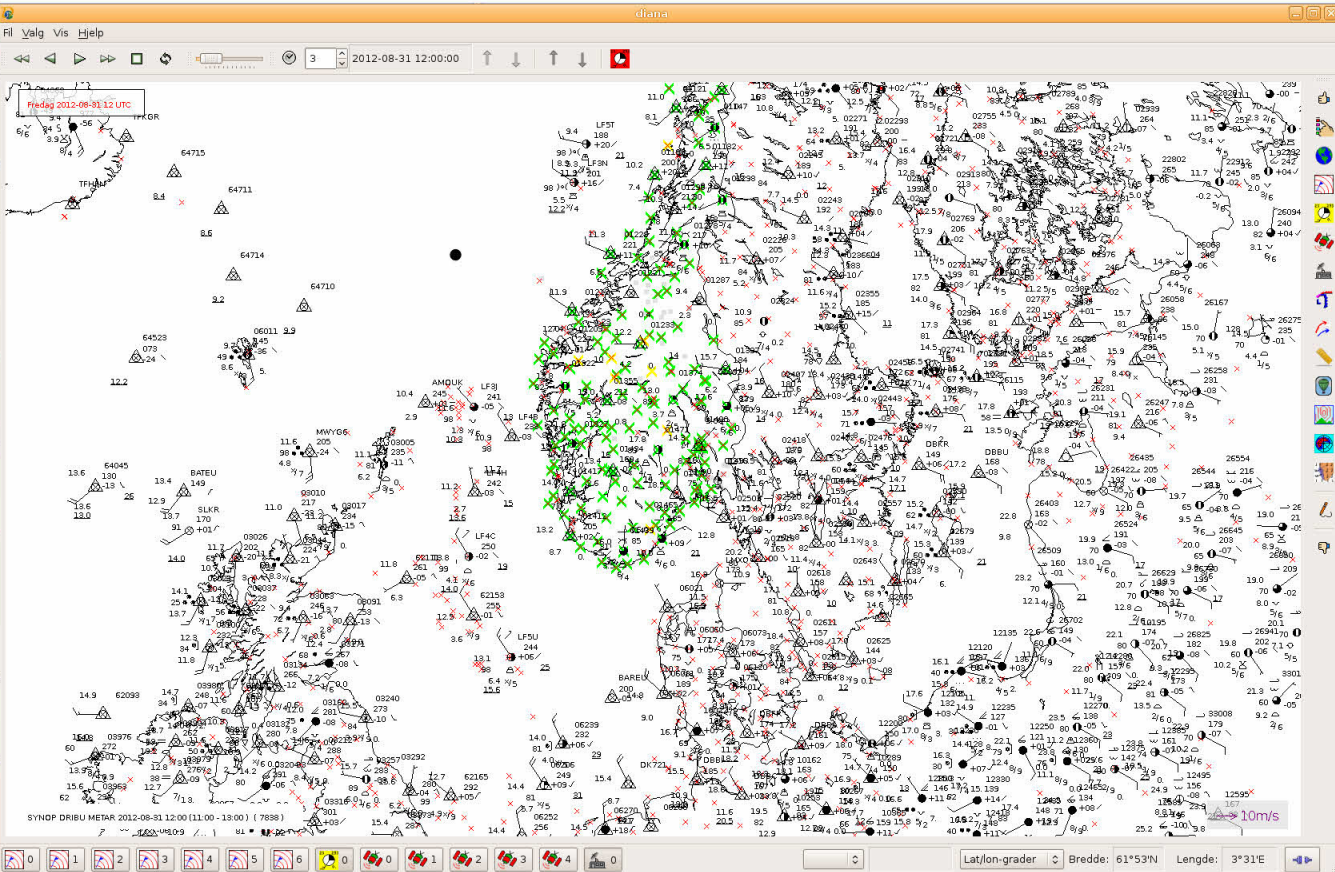


Figure 2B. Example of Stinfosys metadata as displayed for the forecaster

----- Generell informasjon -----

Stasjons skjema

Stasjonsnavn	KONSMO - HØYLAND	Breddegrad	58° 16, 02' N	
Nasjonalt stasjonsnummer	41670	Langdegrad	7° 22, 84' Ø	
WMO-navn:	KONSMO-HOYLAND	WMO-nr	1439	
Kortnavn	Konsmo	Idrift fra:	1992-01-01	
Høyder	Hs = 263	-----	-----	
Kommune	AUDNEDAL	Fylke	VEST-AGDER	
WMO VS		Årsmiddel Fukt/Temp:	-----	

Hovedgrupper av værelementer som observeres:

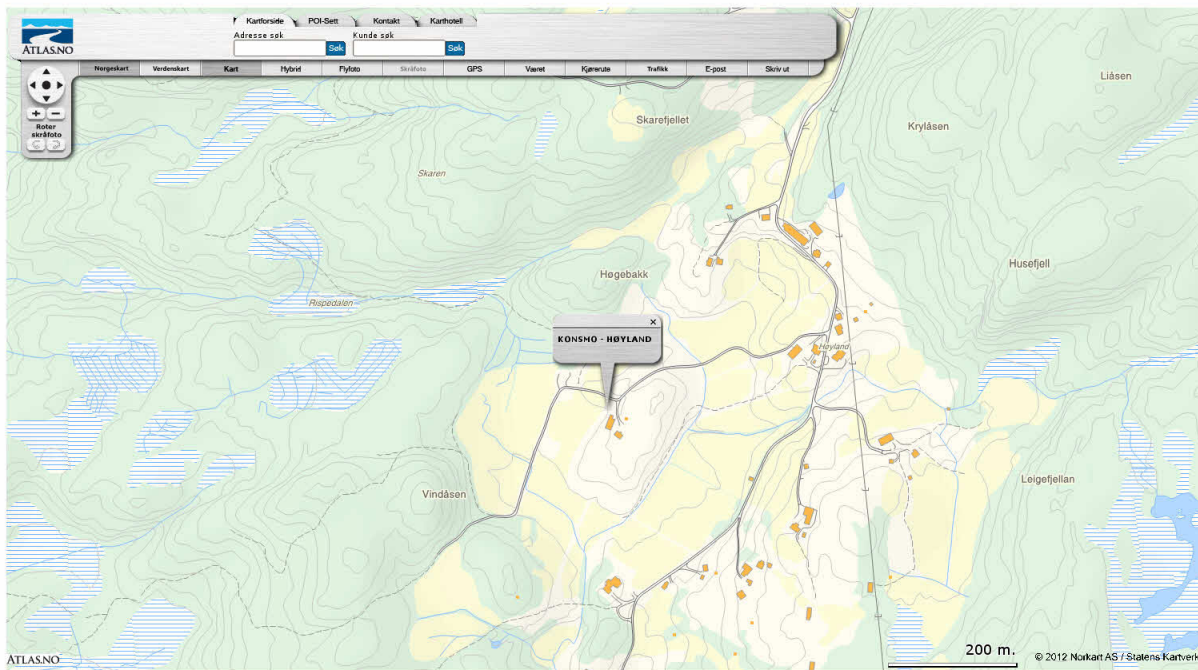
Manuelle målinger: -Temp, -Fukt, -Vind, -Nedbør, -Snødybde, -Skyer, -Vær, -Sikt.

Stasjonsholder: MET.NO

Adresse: POSTBOKS 43 BLINDERN, 0313 OSLO

Nabostasjoner:

41200 FINSLAND 14 km mot NØ (aktiv siden 1971) 41840 HEGEBOSTAD - STAKKSTEINSLIKNUTEN 16 km mot NV (aktiv siden 2000) 41795 OSESTAD 20 km mot SV (aktiv siden 2012) 41860 KVINESHEI - SØRHELLE 24 km mot V (aktiv siden 1995) 41820 KVÅVIK 25 km mot SV (aktiv siden 1972) 41825 LYNGDAL 25 km mot SV (aktiv siden 2005) 41005 SUVATNET 25 km mot SØ (aktiv siden 2012) 39450 DALEHEFFE 26 km mot Ø (aktiv siden 2012) 41090 MANDAL III 27 km mot S (aktiv siden 2009) 39220 HESTAD I ODDERNES 30 km mot Ø (aktiv siden 1900) 42490 EIEISLAND 32 km mot NV (aktiv siden 2012)



----- Teknisk utstyr: -----

KONSMO - HØYLAND, 41670

Teknisk beskrivelse

Sensorbestykning og montasjedetaljer


Modellnavn	Type	Seriernr	Fra	Modell første gang på stasjon	Fysisk høyde	Distribusjon
TESTO 625 (Fuktighetsmåler)	fuktmåler	01845442 - 01902680	2011-10-19	2008-09-08	2	fritt
NEDBØRMÅLER, NORSK	nedbørmåler	..	1991-10-28	1991-10-28	2	fritt
TERMOMETER MIN	temperaturmåler	528/99	1999-09-23	1999-08-18	2.18	fritt
TERMOMETER HOVED	temperaturmåler	1909/97	2000-07-13	2000-07-13	2.18	fritt
TERMOMETER MAKS	temperaturmåler	295/99	2000-07-13	2000-07-13	2.18	fritt
TERMOMETER HOVED	temperaturmåler	805340	2011-10-19	2000-07-13	2	fritt
TERMOMETER MAKS	temperaturmåler	7720/68	2011-10-19	2000-07-13	2	fritt

Figure 2C. Example of Stinfosys station photos


-----Velg stasjon----- [Tilbake til Startsiden](#) **KONSMO - HØYLAND, 41670** [Alle år](#)

Stasjonsbeskrivelser; kart/skisser/bilder

Instrumentoppstilling



Panorama



Bilder

