FINNISH METEOROLOGICAL INSTITUTE

QC1

Automatic Quality Control Phase 1

Plan for the New Quality Control

Kari Niemelä

Pauli Rissanen

FMI

August 2007

Abbreviations

AWS	Automatic Weather Station
CS	Climatological station
DA	Data Acquisition
DB	Database
FMI	Finnish Meteorological Institute
IT	Information technology
NMS	National Meteorological Service
QC	Quality Control
QC0	Quality Control phase 0
QC1	Quality Control phase 1
QC2	Quality Control phase 2
HQC	Human Quality Control

1. Introduction

This QC method can be applied to any relational database with certain structure and flagging system involved. The program is written in PERL and PostgreSQL database is used. This report concerns mainly FMI's databases and quality systems used. At this moment this model is running at FMI for testing purposes. For general use for any other NMS this program should be adjusted.

This report describes in detail one essential part of quality control flow: quality control phase 1 (QC1). QC1 is responsible for the automatic quality checking of data fed into NMS database. The quality checking is done at the same time as data are inserted into user accessible tables. Together with every value put into database a quality flag is placed into the same data table. This flag tells the end user if the data value is reliable or not. In fact a clearly erroneous value will not be delivered to the end user at all. All errors and suspicious values found in QC1 are reported through logging and reports are also saved into database tables. These reports are checked in HQC (Human Quality Control) and erroneous or suspicious values may be then accepted, corrected, estimated, deleted or maybe replaced by interpolated values.

Concerning the common problematic of meteorological quality control there are some publications that are worth while to be consulted. Actually only one link is needed because this link has a reference list that guides further:

www.smhi.se/hfa_coord/nordklim/task1/Report_HQC_redigert.pdf

This link holds a common report from the Nordic countries on quality control methods. There has been fruitful co-operation between the Nordic countries in the area of quality control methods.

2. Principle of table driven quality control

2.1. Phases of quality control

Quality control of observed meteorological parameters can be divided into four logical phases:
– QC0: quality control at the observing site
– QC1: quality control based on other observation values from the same site
– QC2: quality control based on horizontal test (other stations, analyzes, short forecasts)
– HQC: human quality control

QC0 is made at the observing site at the same as the observation is prepared for sending. At manual stations the observing personnel makes this control based on their experience. Unfortunately humans also may make unintended mistakes for example when typing messages. As techniques have developed these mistakes may sometimes be avoided by automatic control already at the stations. At automatic stations the possible errors are of different type. They are mainly caused by faulty equipment but also by the fact that all sensors cannot cover all situations. The station may have some intelligence to avoid the worst errors but not all such errors that a human observer would notice at once.

QC1 is made at the same as the observation is fed into database. So it is a real time check. When an observation is made there is usually an urgency to get it into use as soon as possible. It has to be checked more or less roughly and this checking phase QC1 makes the fast delivery possible.

QC2 is the checking phase where also observations from neighboring stations are needed. In addition to that also results from numerical analyzes and forecasts may be used. Because of that it cannot be a real time check but made a few (10-20) minutes after the nominal observation time. The methods for QC2 are currently still under development. Therefore it is not included in this project.

HQC is made by human quality controllers based on the results of previous QC-phases. The time scale for HQC is from 30 minutes to several days. On weekends HQC may not be possible but for the most crucial cases. HQC may also be possible afterwards if users have found suspicious cases that have not been found in automatic checks.

2.2. Quality flags

Quality flags are numerical values attached to each individual observational value. They tell which quality check phases the observation has gone through and which result the checks have given. The flag is a four digit number where the least significant digit represents QC0, the tens represent QC1, hundreds QC2 and thousands HQC. For every phase the digit value can vary form 0 to 9, where 0 means that check has not been made.

The principle in the use of flags is that the smaller (if not zero) the flag value is in the most significant digit the more reliable the observed value is. Also the more quality check phases the observation has gone through the more reliable the value is. In the following table are listed the flag values that are in use at NMS.

Table 1. Flag values in use at NMS.

Check result or status	Digit
No check made	0
OK	1
Corrected	2
Calculated	3
Interpolated (spatial)	4
Suspected, small difference	5
Suspected, big difference	6
(Digit not in use)	7
Missing	8
Deleted	9

2.3. Table driven quality control

By definition QC1 is the phase of quality control where observed values are checked against fixed limits or other parameter values from the same observing station.

At NMS QC1 (Quality Control 1) is an essential phase of data processing. At the same as it transfers data into ordinary database tables it performs checks on the data it handles. The parser processes receive data from the observing system and write it into table RAWDATA. QC1 checks this table by certain intervals and reads into the process all data where a certain parameter TRANSFER is NULL. This data is handled one station and observation time at a time. After the data is written into ordinary (user accessible) tables the transfer code is set to 1. Quality check is performed on the observation set that is handled together. Some comparison is also made on recent values (previous observations) already in database.

The model of table driven QC is strongly parameterized by using database tables to contain limits and rules to control quality of data stored into database. For most parameters this approach is sufficient and so there are very few if any methods needed that are attached to one specific parameter. This method also allows simple and fast changes into the limits applied to any parameter.

The basic table concerning database use and QC is named **PARAMETER**. Each parameter to be stored into database by QC1-program must be defined in this table. In this table also every parameter that will go through QC has a column named CQ1_METHOD which defines what specific procedures will be used to control the quality of each individual value to be stored into the database. For some parameters it is possible to leave the column CQ1_METHOD null. In such a case no quality control is performed. Typically this kind of parameters are derived parameters or parameters representing mean or average values. For these the quality check is included in the calculation method.

Quality control implies that together with a data value a flag is saved into the database. The flag is an indicator telling which phases of QC a data item has gone through and how useful the value is. An erroneous value is not allowed to be forwarded to any end user.

2.4. Simple one parameter tests

The process of QC1 starts with simple tests ensuring that impossible values are flagged erroneous. Each row in table **QC1_LIMIT** describes a specific checking method used for one or several parameters. This method will be applied to every parameter having the method name attached to it in table PARAMETER. Table QC1_LIMIT contains the universal absolute extremes for this method (never allowed to be exceeded). In principle these limits represent impossible values.

Table **QC1_MONLIM** contains the monthly (upper and lower) limits of each method. The values in this table are typically climatological records for each month or values near those (maybe at 95 % probability limit). If column OVERRUNGRAD is not null exceeding this limit is allowed if step (per 3 hours) from previous observation is smaller than the value given in this column. The purpose of this is the fact that record values are usually reached in a slow process and passing the limit may be accepted if the change from the previous value is sufficiently small.

2.5. Step checks

If the method associated with a certain parameter is found in table **QC1_STEPCHECK** hourly change check is performed. The value is compared to previous value of this parameter on same observing site. Table QC1_STEPCHECK contains the largest allowed hourly change for this method. If the time difference to the previous observation is larger than 3 hours it is not reasonable to do this test. The allowed change is given as 3 hour interval. If there is a shorter time to the previous observation the difference is adjusted so that for 10 minutes' interval the allowed step is about one third of 3 hour difference.

2.6. Persistency checks

Persistency check is useful mainly for wind direction measurements. In winter it may happen that a mechanical wind direction sensor gets frozen. In such a case its output stays the same. This can be tested in persistency check.

2.7. Consistency checks

Consistency checks are checks that are performed between two or more parameters. Mostly in these checks present weather values are compared to visibility or temperature. In some countries the short code (90…99) is mostly used for visibility coding, but it is satisfactorily sufficient for most cases.

There is also a method to check extreme values on a certain time period against individual values of corresponding parameter observed during this period. For example if maximum temperature is observed during 12 hours no individual observation should be larger than the value measured as maximum temperature for this period. A small difference may be allowed due to measuring accuracy.

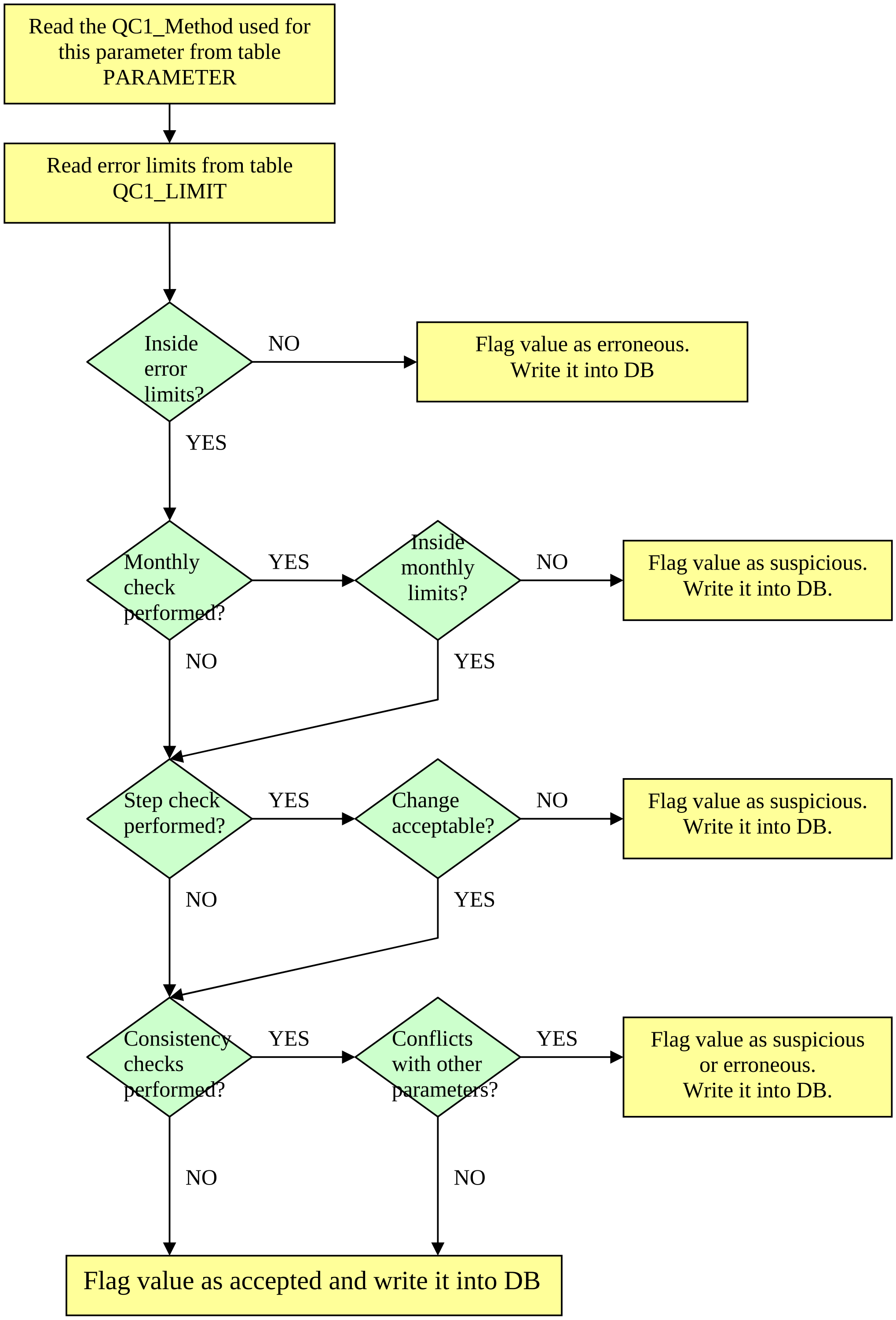

Figure 1. Simplified flow chart for QC1-check of an individual parameter

3. Table structure of NMS QC-tables

NMS database contains in principle two kinds of tables: data tables and metadata tables. The data tables contain the observational data and metadata tables contain definitions that control data storing and for example quality check. The basic metadata table is S_REG (station register) that contains information about the observing stations. Its structure is discussed in a different presentation.

There are about ten data tables where observational data is stored: WEATHER, PREC, PREC_INT, PHENO, etc. They contain identification columns such as PINN (station number) and OBSTIME and a varying number of data columns. For every data column there is another column containing the quality flag value resulting from various checks performed on the observation.

3.1. Table Parameter

This table contains columns:

```
PARM_NAME      VARCHAR (20) PRIMARY KEY,
PARM_ID        NUMERIC (10),
BUFR_ID        NUMERIC (6) or VARCHAR (6),
UNIT           VARCHAR (20),
QC1_METHOD     VARCHAR (20),
DESTINATION    VARCHAR (20),
REMARK         VARCHAR (100)
```

Every observational parameter that should be written into database by automatic processes **must be** found in table PARAMETER. If quality check is made the column QC1_METHOD must contain the name for the method used. Column DESTINATION contains the name of the data table where the parameter value is written. The destination table must contain the column named exactly like PARM_NAME.

A simple example: Four rows from table PARAMETER:

PARM_NAME	PARM_ID	BUFR_ID	UNIT	QC1_METHOD	DESTINATION	REMARK
T	Null	Null	C	T	WEATHER	...
WW_MAN	Null	Null	Code	WW_MAN	WEATHER	…
RR_PER	Null	Null	mm	RR_PER	PREC	…
WD_10MIN	Null	Null	deg	WD_10MIN	WEATHER	…

In the first approach it was planned that QC1 method could be independent from parameter and so a common method could have been used for several suitable parameters. The functionality of HQC-software has, however, the demand that the exact parameter name can be linked to every error message. The QC1-software still allows this flexibility.

The first row instructs the database feeding programs to check every value that is stored into database as parameter T to be checked according to rules given by method T and then written together with flag value into database table WEATHER. The method T is designed for temperature observed near ground, e.g. 2.m. It contains at first a coarse check that the temperature is inside limits that are impossible values. Secondly more sophisticated limits are checked on monthly basis and then the value is compared with the previous observation from the same station.

The second row contains the definition for human present weather observation. Check according to method WW_MAN is performed. The allowed values for this observation are between 0 and 99. In addition to that some checks are made where the observed value is compared with temperature and visibility.

The third example is for precipitation. For it only coarse check and monthly check are relevant. The checked observation value is written into table PREC. The fourth row is for normal synoptic wind speed.

3.2. Basic limit check and table QC1_LIMIT

Each QC1_Method found in table PARAMETER must have a respective row in table QC1_LIMIT. This table has columns:

```
QC1_METHOD     VARCHAR(50) PRIMARY KEY,
GENMIN         NUMERIC (7,1),
GENMAX         NUMERIC (7,1),
SPECVAL        NUMERIC (7,1)
SEVERITY       VARCHAR(50) NOT NULL,
SHORT_MSG      VARCHAR(50) NOT NULL
```

As a key there must be the same method name as in table PARAMETER. The two following columns contain the absolute minimum and maximum values that the parameters presented by the method are **under no circumstances** expected to reach. For the three cases discussed previously there are following rows in table QC1_LIMIT:

QC1_METHOD	GENMIN	GENMAX	SPECVAL	SEVERITY	SHORT _MSG
T	-40	60	Null	ERROR	QC1_GEN_T
WW_MAN	0	99	Null	ERROR	QC1_GEN_WWMAN
RR_PER	0	100	Null	ERROR	QC1_GEN_RR
WD_10MIN	0	360	999	ERROR	QC1_GEN_WD

The method TEMP_LOW used for 2 meter temperature contains minimum value of -40 degrees and maximum value 60. In principle the values found in these columns should be a little more extreme as the all time records ever observed in the country. The excess depends also on the parameter. Maybe for precipitation the excess might be a bit larger than for temperature. For parameters presented as code (e.g. present weather and total cloudiness) the minimum and maximum values are known exactly.

There is also a column named SPECVAL. It represents an individual data value to be allowed outside the range given by GENMIN and GENMAX. It is for the time being used only for method WD_10MIN. NMS uses value 999 to indicate variable wind. To make error checking reasonable this value has also to be allowed.

If a parameter value is outside these absolute limits an error flag is written into the data table and an error message is written into table QC_REMARKS. There it can be found by the person performing manual error checking with HQC-program.

The minimum and maximum values that are in this table or any other QC1-table are not meant to be definitive. They may be adjusted by the persons who perform manual error checking.

Besides basic limit check also the data tables may restrict data values written into database. All databases have an internal characteristic called CONSTRAINT which limits the lower and/or upper limit of the value. These constraints can be defined for each column. For most of the data parameters there are such constraints defined. If an observation value is outside its constraint limit it cannot be written into database. In such a case the value is flagged as missing and an error message is issued.

3.3. Monthly limit check and table QC1_MONLIM

If an observation passes the previous check a second check is made if appropriate. This monthly check only suits a limited number of parameters. The most suitable are temperature and precipitation that are clearly dependent on season. Such parameters are also sunshine duration and snow depth.

The table QC1_MONLIM has the following structure:

```
QC1_METHOD      VARCHAR(50) NOT NULL,
MONTH           INTEGER NOT NULL,
MIN             NUMERIC (7,1),
MAX             NUMERIC (7,1),
OVERRUNGRAD     NUMERIC (7,1),
SEVERITY        VARCHAR(50) NOT NULL,
SHORT_MSG       VARCHAR(50) NOT NULL,
PRIMARY KEY (QC1_METHOD, MONTH)
```

If we take again a few examples this method is easier to comprehend:

QC1_METHOD	MONTH	MIN	MAX	OVERRUNGRAD	SEVERITY	SHORT_MSG
T	1	-35	10	1	SUSPICION	QC1_MONLIM_T
T	6	-5	37	1	SUSPICION	QC1_MONLIM_T
RR_PER	1	0	10	Null	SUSPICION	QC1_MONLIM_PREC
RR_PER	6	0	100	Null	SUSPICION	QC1_MONLIM_PREC

In previous table there are as an example two rows for method T and two rows for method RR_PER. The values should be adjusted so that very seldom but, however, in some rare occasions there is the possibility to reach these limits. There is also a column called OVERRUNGRAD. It bases on the fact that extreme values are usually reached very slowly. So if the extreme value is of the same order than the previous observation the value may be accepted. OVERRUNGRAD represents the accepted maximum deviation from the previous observation when an extreme value is reached. For precipitation this overrun possibility is not relevant and so the OVERRUNGRAD is null (has no value) and exceeding of the limits is never accepted.

3.4. Step check and table QC1_STEPCHECK

Step check is made to check the change the parameter has experienced at the same station since previous observation. This test is not relevant for many parameters but for temperature and related parameters it can be used.

Structure of the table QC1_STEPCHECK is as follows:

QC1_METHOD VARCHAR(50) PRIMARY KEY,
INCRLIM NUMERIC(7,1),
DECRLIM NUMERIC(7,1),
SEVERITY VARCHAR(50) NOT NULL,
SHORT_MSG VARCHAR(50) NOT NULL

A few rows as an example:

QC1_METHOD	INCRLIM	DECRLIM	SEVERITY	SHORT_MSG
T	7	7	SUSPICION	QC1_STEP_T
TD	5	5	SUSPICION	QC1_STEP_TD
P_STAT	5	12	SUSPICION	QC1_STEP_P

INCRLIM represents the allowed change when the current observation value is larger than the previous one and DECRLIM gives the allowed change when the current observation value is smaller than the previous one. At least for pressure a larger limit has to be used for decreasing situations.

3.5. Persistency checks

In persistency check there is a check made that the parameter value has changed from the previous of a few previous observations. Maybe the only relevant parameter for which this check should be made is wind direction. In winter circumstances a mechanical wind direction sensor may get frozen and give the same value until it has melted.

The table structure for QC1_PERSISTENCY is as follows:

QC1_METHOD VARCHAR(20) PRIMARY KEY,
HOURS NUMERIC(2),
SEVERITY VARCHAR(20) NOT NULL,
SHORT _MSG VARCHAR(50) NOT NULL

The column HOURS gives the time interval during which the persistency is observed. As an example values for wind direction are given. So according to these values, if wind direction has stayed the same for 12 hours an error message will be issued.

QC1_METHOD	HOURS	SEVERITY	SHORT_MSG
WD_10MIN	12	SUSPICION	QC1_PERS_WD

3.6. Consistency checks

Naming 'consistency check' is used for quality checks where an observation value is compared with the value of another observed meteorological parameter. Maybe the simplest example is that dew point has to be smaller or at the most the same as temperature. At automatic stations this kind of checks are made in the processing unit, but at manual stations a typing error may cause such an error to happen.

In the system that is working at NMS there are two different types of consistency checks in use:
– consistency (pure consistency)
– compare

Pure consistency checks are mainly performed between present weather and temperature, humidity or visibility. The structure of the table QC1_CONSISTENCY is following:

```
QC1_METHOD     VARCHAR(20) NOT NULL,
PAR_VAL        NUMERIC(7,1) NOT NULL,
COMP_PAR       VARCHAR(20) NOT NULL,
CPAR_MIN       NUMERIC(7,1),
CPAR_MAX       NUMERIC(7,1),
SEVERITY       VARCHAR(50) NOT NULL,
SHORT_MSG      VARCHAR(50) NOT NULL,
PRIMARY KEY (QC1_Method, PAR_VAL, COMP_PAR)
```

When an observation value for a parameter checked according to a certain method is found to be a PAR_VAL it is checked against parameter defined as COMP_PAR and the value of COMP_PAR has to be larger than CPAR_MIN and smaller than CPAR_MAX. One of the compared values may also be null (not given) and then only one comparison is made.

The next examples give a better view of this feature:

QC1_METHOD	PAR_VAL	COMP_PAR	CPAR_MIN	CPAR_MAX	SEVERITY	SHORT_MSG
WW_MAN	10	VIS_MAN	1000	10000	ERROR	QC1_WW_VIS_10
WW_MAN	42	VIS_MAN	Null	1000	ERROR	QC1_WW_VIS_42
WW_MAN	51	T	-5	Null	SUSPICION	QC1_WW_T_51
WW_MAN	42	RH	90	100	SUSPICION	QC1_WW_RH_42
WW_AWS	34	VIS_AWS	Null	1000	ERROR	QC1_WAWA_VIS_32

The first row gives the rule to check ww-code 10 against visibility. Visibility must be between 1 and 10 kilometers. If the matter is not so an error will be issued. Either ww-code or visibility is faulty and the human quality control has to decide which one.

Mostly only “SUSPICION” is the severity that can be “doomed”. Some strict rules, however, allow also “ERROR” to be issued.

The other table controlling relations between parameters is QC1_COMPARE. It is defined as follows:

```
QC1_METHOD      VARCHAR(50) NOT NULL,
RELATION        VARCHAR(10),
COMP_PAR        VARCHAR(50),
TIMESCALE       NUMERIC(7,1),
TOLERANCE       NUMERIC(7,1),
SEVERITY        VARCHAR(20) NOT NULL,
SHORT_MSG       VARCHAR(20) NOT NULL,
PRIMARY KEY (QC1_METHOD, COMP_PAR)
```

Here a strict comparison between two parameter values is made. There is a possibility to also add a time scale for maximum and minimum temperature observations.

Possible relations are <, >, <=, >=, >> and << but these can also be substituted by their literal forms lt, gt, le, ge, max and min. These relations are interpreted so that the value of the parameter studied has to fulfill the relation towards the values of COMP_PAR during the period of TIMESCALE hours. The last two expressions are a bit more complicated.

Again some examples:

QC1_METHOD	RELATION	COMP_PAR	TIMESCALE	TOLERANCE	SEVERITY	SHORT_MSG
TMAX_12H	>=	T	12	0.2	ERROR	QC1_TMAX_T
TMIN_12H	<=	T	12	0.2	ERROR	QC1_TMIN_T
T_0_MIN_12H	<=	TMIN_12H	null	0.5	SUSPICION	QC1_TG_TMIN
TD	<=	T	null	0	ERROR	QC1_TD_T
P_SEA	>	P_STAT	null	0	ERROR	QC1_PRES_MSL_STAT
TMAX_12H	>>	T	12	2	SUSPICION	QC1_TMAX_T
TMIN_12H	<<	T	12	2	SUSPICION	QC1_TMIN_T

For example the observed value of maximum temperature has to be larger than any of temperature observations during measuring period (TIMESCALE). A tolerance of 0.2 degrees is allowed due to observation accuracy. The maximum temperature observation is flagged erroneous, but as well the wrong observation can be one of the temperature values. This will be decided by HQC. The second last row in the example tests that the difference between the maximum value of COMP_PAR (in this case T) and TMAX_12H is not larger than 2 degrees. The last row tests the same for minimum of T and TMIN_12H.

A strict rule is that dew point has to be smaller or equal than temperature. Even no tolerance is allowed because dew point is calculated from temperature. The same counts for air pressure reduced to mean sea level. It always has to be larger than station level pressure.

3.7. Error messages

In every QC1-table there is a column named SHORT_MSG. It contains a somehow descriptive but short error message. This message can be unique or common to several errors. It is converted into a longer message with the help of other database tables so that in HQC a plain text message appears into the view of the quality check person.

There are three database tables by which the error messages are controlled: QC_SHORTMSG, QC_MESSAGES and QC_RELATIONS. In principle they are very simple.

QC_SHORTMSG:
ERROR_NO NUMERIC(4) PRIMARY KEY,
SHORT_MSG VARCHAR(50) UNIQUE NOT NULL

This table only gives the conversion from SHORT_MSG into ERROR_NO. A few example rows:

ERROR_NO	**SHORT_MSG**
1003	QC1_GEN_T
1051	QC1_GEN_RR
201	QC1_WW_VIS_0
206	QC1_WW_VIS_42
230	QC1_WW_T_68
303	QC1_MONLIM_RH
304	QC1_MONLIM_SN

QC_MESSAGES:
ERROR_NO NUMERIC(4) PRIMARY KEY,
LANGUAGE VARCHAR(3) NOT NULL,
ERROR_TEXT VARCHAR(256) NOT NULL

Some examples of the error messages:

ERROR_NO	**LANGUAGE**	**ERROR_TEXT**
1003	ENG	General value limit for temperature have been exceeded. This observation is found erroneous.
1051	ENG	General value limits for this parameter have been exceeded. This observation is found erroneous.
201	ENG	A significant ww-code should be reported because visibility is reduced.
206	ENG	Moderate drizzle is reported. Visibility should be less than 4000 m.
230	ENG	Rain and snow has been reported when temperature is below accepted minimum for this weather phenomenon.
303	ENG	Reported relative humidity has passed its monthly limit.
304	ENG	Reported snow depth is above its allowed monthly limit.
304	X	<Error message in X language >

The error messages presented in this table appear as such to the persons doing HQC. One of the columns is LANGUAGE. When the error messages have been translated into defined language they can be used for that purpose. So when the language in HQC is switched into defined language the error messages also appear in defined language.

QC_RELATIONS
ERROR_NO NUMERIC(4) ,
PARAMETER VARCHAR(50)
PRIMARY KEY (ERROR_NO, PARAMETER)

ERROR_NO	**PARAMETER**
1003	T
1051	RR_PER
201	WW_MAN
201	VIS_MAN

This last table gives the link between error number and parameter name to be used in HQC. The same error number may be associated with several parameters because as the result of consistency test one can not tell which of the values is erroneous or suspicious.

3.8. Table QC_REMARKS

All incidents of errors or suspected values found in QC1 are written into table QC_REMARKS. This table is not related only to QC1 but it is appropriate to explain its structure here. It has columns as follows:

PINN NUMERIC(4) NOT NULL,
OBSTIME TIMESTAMP NOT NULL,
ERROR_NO NUMERIC(10) NOT NULL,
ERROR_LEVEL NUMERIC(4),
REMARK_DATA VARCHAR(150),
CREATED_ON TIMESTAMP NOT NULL DEFAULT NOW(),
DB_USER VARCHAR(20),
CTRL_DATE TIMESTAMP,
QC_LEVEL NUMERIC(2),
PRIMARY KEY (PINN, OBSTIME, ERROR_NO, CREATED_ON)

Part of the columns like PINN, OBSTIME and ERROR_NO are self-explanatory. Some need a few words of description. ERROR_LEVEL is a column by which error messages may be grouped. REMARK_DATA contains the erroneous or suspicious data value and some other descriptive information written by QC1. CREATED_ON holds the timestamp when the check was made. DB_USER is reserved for HQC and it contains the user name of the person doing HQC. CRTL_DATE contains the timestamp of the HQC. QC_LEVEL holds simply the phase of QC that has written the remark into this table.

4. Maintenance of QC1-tables

The limits originally written into QC1-tables by FMI-experts are “first guesses” to make it possible to test the program functionality. It is meant that these limits are reconsidered and adjusted by the personnel of NMS and especially the ones that will be involved with HQC. It is obvious that some tests are in the beginning too strict and others do not produce errors at all. With some experience in the practice (concerning the monthly limits this may take even years) suitable limits will be found. Doing these changes is not difficult but it needs some training and always when a change is made certain care has to be taken.

In practice the change in a desired value is made by changing the value in the correct row and column in the appropriate QC1-table. What will be the method used to do that is not clear yet. SQL-language is one way, but hopefully a more comfortable user interface is possible.

When a new test is added into a QC1-table care has to be taken that also possible new error messages are written into proper tables. So if a new SHORT_MSG is added into one of the tables a new row has to be added into tables QC_SHORTMSG (containing the pointer to ERROR_NO) and QC_MESSAGES (containing the ERROR_TEXT).

5. The QC1-software

The QC1-software is the engine that does the job round the QC1-tables. By default it reads from RAWDATA-table **all** rows where TRANSFER-code is null or zero. In special cases also time parameters can be used to control its behavior.

The software reads data and groups it internally so that all parameter values to be written into same row in the database are treated at the same (station/time/height/destination). So the error checks made in QC1 are restricted into such a union. The most usual case is such that the parameters from a synop-message from one station are treated together and the error checks are made. There may be data coming from another source and at a different time. By present methods these cannot be checked against values already in the database. Of course there is nothing that prevents NMS to develop new methods for error checking in the future.

The programming has been made with Perl-language. It is a script language but it allows structural methods. This is also used in QC1-program. It consists of a short main-module and several subroutines. All modules are in the same file qc1.pl.

5.1. Restrictions

The date and time operations inside qc1.pl are programmed with the assumption that date format in the database is according to ISO-8601 convention. This means that date/time format is

```
yyyy-mm-dd hh:mm:ss
```

There is no guarantee of proper functioning of the program if other date format is used. In fact if other format is going to be used the program has to be reprogrammed by the parts dealing with dates and times.

5.2. Calling the QC1-program

Normally the QC1-program runs in a few minutes (1 to 5) intervals controlled by a shell script. It can be called also on a command line in special occasions. If it is called without any parameters it will handle all data rows in RAWDATA-table where TRANSFER-column has a value of null or zero. It can also be called with a time parameter and in such a case it will handle all data that fits the time definition with wildcards appended on both ends. Even then the program will not handle data where TRANSFER-code is set.

Calling procedure:

```
qc1.pl [yyyy-][mm-][dd] [hh:][mm:][ss]
```

Any time string appended on the command line will be inserted as such (without any reasonability checks) into the retrieval request which collects data from RAWDATA-table. If a string 06-06-06 is appended the program tries to find all data from 6th of June 2006. If 06:06:06 is appended the program tries to find data from all observations where hours, minutes and seconds are 6 but there hardly is such data in database. A time parameter which is not suitable for the time string will result only in that no data will be found from RAWDATA-table. After all, in routine use there is not much benefit from this time control feature.

5.3. Output

The program writes as output every insert and update clause it performs. It also writes out every clause it writes into QC_REMARKS. It is advisable to redirect the output into some logfile and save it for some days for possible manual inspection.

The redirection is made in unix-environment by one or two >-marks. If only one mark is used every new command destroys the possible existing file by the same name. If two marks are used new output is appended at the end of the file. The output file can be renamed once a day and so it will not grow too large. The error messages are redirected with number 2 and the same marks. The command could thus be:

```
qc1.pl >> outputfilename 2>> errorfilename
```

5.4. Structure and functionality of the QC1-program

The QC1-program consists of a short main module and – when this is written – 19 subroutines. The number of subroutines will increase with a few when the structure of the system gets its final state.

The subroutines can be divided into a few groups according to their functionality. There are error check routines, insert routines and assistance routines.

Error check routines:
– check_error – overcoat to other error check routines
– check_error_limit – basic limit check
– check_monthly_limit – monthly limit check
– step_check
– persistency_check
– consistency_check
– compare_check

The names of these subroutines tell the nature of these routines. The first one acts as an overcoat to other routines. These routines have as input parameters all the information they need to know to make the error check. They read from the database the definitions for the error check. If an error is found they call the routine report_error but otherwise they set the flag value to good.

Insert routines:
– insert_data – overcoat to other insert routines
– insert_weather – insert into table WEATHER
– insert_prec – insert into table PREC
– insert_daily – insert into table DAILY

– insert_univ – universal insert into data tables

The first of these routines, insert_data, holds the main functionality of this program, it sorts the data it receives after get_rawdata is called, and calls also the quality check routines. After that it calls the specific insert routines. The insert routines are named according to the table in which they are going to make the data insert. Until now there are three specific insert routines. The universal routine insert_univ is able to take care of some of the remaining tables but at least for soundings two insert routines have to be made yet.

Assistance routines

– get_rawdata – read data from table RAWDATA
– check_if_insert – check if data should be written with insert or with update
– check_if_insert_d – check if data should be written with insert or with update
– check_if_insert_p – check if data should be written with insert or with update
– check_if_synop – check if synop time
– get_qc_method – read basic info from table PARAMETER
– report_error – write error report into table QC_REMARKS

The routines in this category make different small assistance duties. The first one, get_rawdata, reads in the start of the program all the relevant data into the memory of the program. The next three check if the data have to be written into database by insert-clause or by update-clause. There can be only one row in the database for each station and time (concerning soundings and mast data also height). When such a combination exists already in the database an update has to be made. When a new row is added an insert has to be made.

The table WEATHER has a column SYNOP, where there is the information if the observation is at synop-time. At the present this info is determined by function check_if_synop only by the fact that the observation is made at an even hour divisible evenly by 3. At a later stage a more sophisticated function may be needed.

The routine get_qc_method reads from table PARAMETER the two basic facts needed:
– which error checking method each parameter is using (if any)
– into which table the parameter value should be written
If there is no entry in table PARAMETER for some parameter QC1 ignores the parameter (leaves outside every treatment). No error message is issued in such a situation.

The routine report_error is called if error checking has found something erroneous or suspicious in some parameter value. A row is written into table QC_REMARKS. Also a remark will be written into the output of the program.