Internal Memo

Title: Hiding entries in the qc1 database

From: **Reinhard Hanuschik**
To: **DFO team**
Date: **2003-01-30**
Purpose: **Requirements for QC1 database structure**
and for an interactive tool to hide entries (qc1Hide)

TBD: **Modifying timestamps

Change record:

version 1.0	2003-02-14

1 Motivation

1.1 QC1 database: functions overwrite and hide

Data are entered by DFO into the QC1 database using qc1Ingest. This tool is called from a script or from a command line.

It may become relevant to update an existing entry, either for overwriting or for hiding.

Overwrite. It may happen anytime that a record needs to be inserted which either would duplicate an existing one, or would modify it. For operational reasons, such cases should be handled in an automatic and transparent way. The procedure should be that existing entries are recognized by the database infrastructure and flagged. Technically, this should be handled as hiding the old entry.

Hiding. There may also be cases that after ingestion it is recognized that an entry is invalid. This corresponds to hiding without replacing the record ('active hiding'). Two operationally relevant use cases are:

- Invalidation of a created calibration product.
- Ingestion of a data set for a non-standard set-up.

Hiding is an interactive and iterative process. First we need to find the record, and then we want to hide it.

1.2 Functions and tools

Both functions (overwriting and hiding) require an extension of the existing database structure.

The function 'overwrite' can be provided in an automatic way by a database trigger. The function 'hide' should be supported by an interactive tool. This tool will be called qc1Hide in the following.

Entries to hide could always be deleted from the database by using SQL commands. However, there are two reasons for making it reasonable to have a tool assisting in this job. One is that manipulating the database with SQL commands requires a certain level of expertise. An interactive tool facilitates such manipulations in a non-expert mode. The second one is that the policy for dealing with QC1 database entries is to hide rather than delete, and this requires more complex database manipulations.

1.3 Deletion vs. hiding

True deletion will always be possible at the SQL level, but it is not actively supported by qc1Hide. Deletion should be reserved to technical interventions, when e.g. a set of BIAS QC1 data by mistake made it into the FLAT table. These data should be *deleted* from the FLAT table. Nevertheless, the more common case is that a set of QC1 data has been reprocessed. Then this one should become the valid one, while the old one needs to be hidden.

2 Requirements on database structure

To provide proper management of QC1 database entries, each record of the QC1 tables needs to have two new keys called 'visib_flag' and 'timestamp', in addition to the keys already defined

2.1 visib_flag

The `visib_flag` controls the visibility of a record. It has either the value 'N' or 'Y', default being 'Y' (record visible). All records having value 'N' are *not* visible to any of the functions of the qc1Browser or the qc1Plotter. I.e., they don't show up, they don't plot, they are not evaluated for statistics, and they are not used for any of the filter options.

The `visib_flag` becomes relevant only when records are either updated or actively hidden. New records are always visible. When an existing record is updated, effectively the old one gets `visib_flag` = 'N' and the new one gets `visib_flag` = 'Y'. When an existing record is hidden, its `visib_flag` is set to 'N'. There is no deletion in this concept, all invalid entries are just being hidden.

2.2 Timestamp

Each QC1 table has a '`timestamp`' column. A new record receives an ingestion timestamp. On any follow-up manipulation, i.e. hiding, a new record is inserted, again with a timestamp. This procedure records the full history of a data set.

Modifying timestamps of existing entries means that we are **not able to recover the complete history, although it makes sense. So is this what we want? The example below has a different scheme: timestamps of existing entries are not modified.**

2.3 Example

A new record is inserted (modified parts are shaded):

timestamp	time and filter keys = primary key				QC1 parameter values		visib_flag
	mjd_obs	bin	arm	etc.	value1	value2…	
2003-01-31T12:31:00	52331.489	1x1	red	…	1.332	1.234	Y

This record is updated later:

timestamp	time and filter keys = primary key				QC1 parameter values		visib_flag
	mjd_obs	bin	arm	etc.	value1	value2…	
2003-01-31T12:31:00	52331.489	1x1	red	…	1.332	1.234	N
2003-02-05T17:21:34	52331.489	1x1	red	…	1.335	1.234	Y

3 Trigger

A database procedure is needed to ensure that for any record all n-1 entries have `visib_flag` = 'N', and only the latest one has 'Y'. On ingestion of a new record, this procedure has to

- insert the timestamp and `visib_flag` = Y for that record;
- check if there is an older record with the same primary key which would then get `visib_flag` = 'N'.

That procedure should become part of `qc1Ingest`.

4 Primary keys

In order to identify a record, we need to define a primary key. We propose to use a composite primary key with the following components:

- `mjd_obs`
- a set of instrument keywords (those with the filter flag set)
- `visib_flag`

The instrument keywords define logically identical basic data sets. They correspond to the keywords used e.g. for master calibration renaming or for association. In the context of association, they are called match keys. In the framework of the calibration archive, they define validity chains.

The `mjd_obs` information is needed to identify a certain record within the basic data sets.

The `visib_flag` is used in case there are multiple hits for the (mjd_obs/instrument keys) combination. Only one of them is allowed to have `visib_flag` = Y.

5 qc1Hide: the interactive tool for hiding

Apart from the automatic process to hide an older record version if a new one is ingested, an interactive tool is needed to select and hide certain records under the user's control. It is called `qc1Hide` in the following.

It is anticipated that two ways of selection are useful:

- select by instrument parameters
- select by date.

In either case it is proposed to have `qc1Hide` as a web-based form.

5.1 Select by parameters

This selection process has four steps (Figure 1):

1. select the instrument
2. select the database table
3. select the instrument parameters
4. select one or more records for hiding.

For each of these steps, a web page is needed which has a standard content and can be created on the fly by a script reading configuration files.

Alternatively, a Java-script could be used to create the complete query form dynamically.

5.1.1 Instrument selection

A database table is read to offer the supported instruments. The proposed web interface is shown in Figure 2.
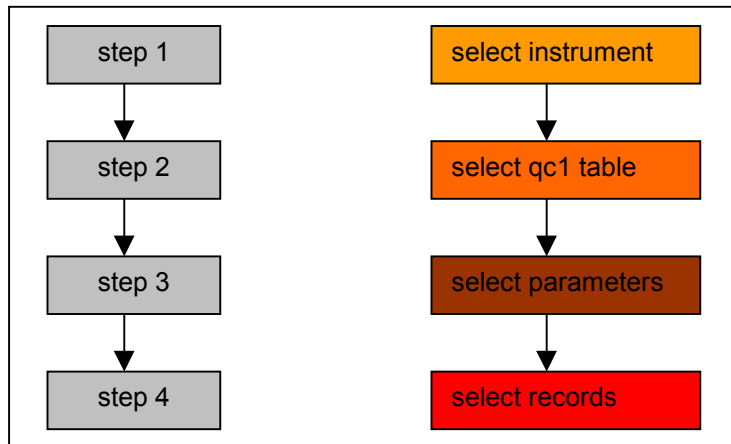
Figure 1: The parameter selection process

5.1.2   Database table selection

A database table is read to offer the QC1 tables available for the selected instrument. Their entries are read from the database configuration tables. The web interface is shown in Figure 3.

5.1.3   Instrument parameter selection

Next, the instrument parameters have to be selected. Their names are read from the existing configuration file. Only those marked by the filter flag should be offered here. The user selects for each of them a value. The options are read from the database entries (`select distinct`), in the same way as for the `qc1Browser` and `qc1Plotter`.

Selection values are passed through a form to a cgi script with the database query (Figure 4).

5.1.4   Record selection

The result of the previous query is transformed into a web page (Figure 5) which shows the records queried. For each of the records, the result page shows

- a select button,
- mjd_obs and civil_date information,
- the entry of 'calibfile'.

`calibfile` is displayed since this is often a fundamental operational entity for DFO. However, it could be NULL since there is no one-to-one relation between `calibfile` and a QC1 record.

The user has the option to select one, several, all or none of the offered records. After the selection is finished, it needs to be confirmed. Then the selection is passed to a database application. That application sets the `visib_flag` for the selected records to 'N'.

5.2   Selection by date

Alternatively a set of records can be selected by date (option 'date' in Figure 2). All data from the QC1 database for the selected instrument and date are offered for selection (Figure 6). The selection process is shorter, but usually the number of hits will be higher in this mode. The advantage is faster multiple selection if e.g. it is known that for some reason all data of a certain day have to be hidden.

6   Security

Since hiding is a database manipulation, the interfaces need to be password-protected. The protection is called within the first interface.

Furthermore, the interface should not be linked from any publicly accessible web page.

Appendix

Figure 2: Interface 1 (Selection of instrument)

qc1Hide: Hide entries of the QC1 database

Instrument : UVES

To find entries to hide, you can either query by **date** or by **parameters**.

If you select by **date**, all entries for the selected date are shown for selection.
If you select by **parameters**, you have to select a specific table. Then you can select entries from that table by parameter values.

Query by date or parameters:	date	parameters
	(YYYY-MM-DD)	select table name
		uves_bias ▾
		uves_bias
		all other uves tables

Submit Reset back

Figure 3: Interface 2 (Selection of database table)

qc1Hide: Hide QC1 entries in uves_bias

Instrument :	UVES
QC1 Database table:	uves_bias
Selection mode:	parameters

Select the parameter values to identify a record. The options for selection are read from the database. You can also choose 'any' (be careful ...).

All hits will be listed with date information.

Select parameter values to be queried:

binning	1x1 ▾
arm	red ▾
	blue
	red
	any

Submit Reset back

Figure 4: Interface 3 (Selection of instrument parameters)

qc1Hide: query results

Instrument :	UVES
QC1 database table:	uves_bias
Selection mode:	parameters

Selected parameter values:	binning:	1x1
	arm:	red

Select record(s) to be hidden:

Hide	civil_date	mjd_obs	associated calib file
☐	2003-01-24	52334.1471123	UV_MBIA_2002-01-24A_1x1bl.fits
☐	2003-01-25	52335.147000	UV_MBIA_2002-01-25A_1x1bl.fits
☐	2003-01-26	52336.1482389	UV_MBIA_2002-01-26A_1x1bl.fits
☐	2003-01-27	52337.1492221	NULL

Hide   Reset   UVES

Figure 5: Interface 4 (Selection of records)

qc1Hide: query results

Instrument :	UVES
Selection mode:	date

Selected date:	2003-01-31

Select record(s) to be hidden:

Hide	mjd_obs	table_name	associated calib file	show all parameters with filter flag set?
☐	52334.14733422	uves_bias	UV_MBIA_2002-01-24A_1x1bl.fits	
☐	52335.14700000	uves_flat	UV_MFLT_2002-01-24A_red5800di1_1x1.fits	
☐	52336.14822134	uves_flat	NULL	
☐	52337.149487324	...	...	

Hide   Reset   back

Figure 6: Selection by date (results)