

Changes in Oracle database July 2008

Ilse Koenig, GSI July 8th, 2008

Table of Contents

1 Overview.....	1
2 New layout for the storage of DAQ files.....	2
2.1 Tables in Oracle account DAQ.....	2
2.2 Interface to store DAQ files in Oracle.....	3
2.3 WebDB interface for DAQ files.....	4
2.4 WebDB interface for the DAQ operator.....	4
3 New and changed user interfaces.....	5
3.1 Beam time logbook.....	5
3.2 Statistic plots for DAQ files.....	5
3.3 Run validation.....	5
3.4 Slow control data.....	6
3.5 Log of HADES magnet and cryo plant.....	6
3.6 Simulation.....	6
3.7 Analysis interface.....	7
3.8 Interface to show and change experiments.....	7

1 Overview

Starting in 2009 the DAQ will use several event builders running in parallel. Each event builder will write its own files. The Oracle table, which stores the DAQ runs taken with one event builder, is not appropriate for the new situation and a new scheme to store the runs/files in the future had to be developed.

The parameter initialization in the analysis code, the Oracle interfaces for parameter and run validation, the various slow control interfaces, the beam time logbook and many more, they all depend on the run id and the corresponding run start.

The change in the storage of DAQ runs therefore requires major changes in various tables, views and packages on several Oracle accounts.

Because many applications needed to be changed anyhow, it was a good opportunity to review the code, to reorganize and clean some accounts (major cleaning was already done in fall 2007) and to rewrite about two dozens of applications, especially WebDB interfaces created before 2002 with the WebDB toolkit.

WebDB is not supported anymore by Oracle since several years and all applications made via the WebDB toolkit are not portable. The new code is much more readable, more flexible, much easier to maintain and would, (with very few changes in some basic packages) also run without the WebDB framework.

To develop and test the new code, all HADES Oracle production accounts have been installed in February 2008 in a test database DB-HADES_TEST (on the same machine as the CBM database, which is actually not used by CBM).

Exports of all Oracle accounts supposed to be changed were generated, as well as HTML reports using TOAD, and stored in the GSI tape robot.

2 New layout for the storage of DAQ files

To minimize the necessary changes and to analyze the files independently, but to guarantee to use the same parameters for the events collected in parallel, it was agreed to use the following scheme:

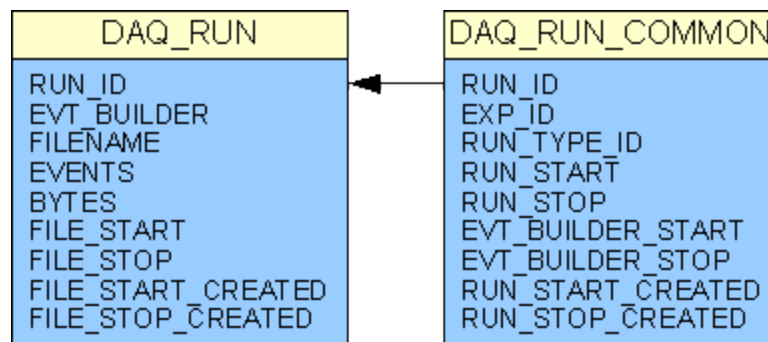
- All files have the same run id calculated from the local time of the master node.
The run id generator used in the past is discarded.
- All file names have the same prefix defining the run type and the same time string.
Therefore the first 13 characters are the same for all files.
- Additionally the file name contains the number of the event builder (01...99).
If only one event builder is used this additional substring can be omitted. This for example is the case for all previous beam times.
- All files start and stop (almost) at the same time. These start and stop times are the individual local times on the different event builders and might slightly differ from the „master“ time distributed via the run id depending on the synchronization of the clocks and the event rate.

The advantage of this scheme is, that the analysis code does not need to be changed. It works for one or several event builders.

But this is only possible, because all necessary changes can be implemented internally in the Oracle tables, views and interfaces.

2.1 Tables in Oracle account DAQ

The run and file information is now stored in two new tables:



The table **DAQ_RUN_COMMON** contains the information common for all files with the same run id, but different event builder numbers.

All files share the same experiment name (`exp_id`) and run type (defined by the 2-character prefix of the files).

The `run_id` is unique (primary key) in this table. The `run_start` and `run_stop` are the times defined by the event builder with the minimum start and stop time. The corresponding number of the event builder is stored in `evt_builder_start`, respectively `evt_builder_stop`.

For debugging also the times when the insert and update was done (`run_start_created` and `run_stop_created`).

The `run_start` must be smaller or equal to the `run_stop` and a run must not have an overlap with the previous and the next run.

The table **DAQ_RUN** contains the list of all files.

The run id and the event builder number form together the primary key.

The filename must be unique and consists of the prefix, the time string and the event builder number (01..99).

For debugging also the start and stop time of the file is stored and also the times the insert and update were done (file_start_created and file_stop_created).

The data are stored in Oracle with PERL scripts running independently on the different event builders.

Each script checks in regular time intervals, if the ASCII file written by the corresponding event builder contains new lines. One line is written at the start of a file and contains the run id, filename, event builder number and file start. An other line is written at the stop of the file and contains the run_id, event builder number, file stop, number of events, file size. In case the event builder crashes, this second line might be missing.

The PERL scripts insert (line written at the start of the file) and update (line written at the stop of a file) the data by calling procedures in a new package **daq_run_mtn** (described below). It takes care of the consistent filling of both tables and checks for overlaps.

Errors are recorded in a separate new table **daq_run_error_log.**

It contains the insert or update parameters, the error message and a flag for the action type.

A WebDB application allows the DAQ operator to check the time difference between the different event builder start and stop times and to check the error log.

The data of all „regular“ existing runs have been copied to the new tables with event builder number 1.
The run type was corrected according to the prefix (if not a typo or other unknown prefix).
The distinction of different daq types has been dropped. (It was anyhow never seriously used.)

Not copied were runs

- with not unique file names (many runs with name test, eb05, xx, ... without the suffix .hld) or NULL or some other nonsense names.
- files written to the NULL device or a disk used only for testing

Because the run id will be in the future calculated from the time, the existing randomly generated run ids cannot be used anymore. By skipping the runs not needed, the risk is lower to match an existing number again in the future.

The **old** table **RUNS** still exists. The data are not lost. But inserts and update privileges have been revoked.

2.2 Interface to store DAQ files in Oracle

Direct changes of the new tables via SQL are restricted to the owner (DAQ) of these tables. Other users must use the new package **daq_run_mtn**. It provides procedures for insert and updates.

Insert:

```
procedure make_insert (  
    p_run_id      in number,    /* run id          */  
    p_event_builder in number,  /* number of the event builder */  
    p_filename    in varchar2,  /* name of the file   */  
    p_file_start  in date );    /* start time of the file */
```

When called, the program checks, if the run exists already in table DAQ_RUN_COMMON.

In case, the run does not exist, it determines the run type from the prefix of the filename and the experiment with the time string of the filename and then fills both tables.

In case, the run exists already in the table DAQ_RUN_COMMON, it first checks, if the start time of this event builder is smaller than the start time already stored in the table. If it is smaller, the new start time and the new event builder number is stored in DAQ_RUN_COMMON. Afterwards the table DAQ_RUN is filled.

At the end, after all PERL scripts have inserted their data for a certain run id, the table DAQ_RUN_COMMON contains the minimum start time of all event builders.

The program also checks, if the stop time of the previous run is smaller than the start time of the new run. Otherwise it changes the stop time of the previous run to the start time of the new run minus one second. This guarantees, that different runs have no time overlap.

Any error (run id exists already, filename not unique) are recorded in the table DAQ_RUN_ERROR_LOG. The “normal” insert is discarded.

Update:

```
procedure make_update (  
    p_run_id      in number, /* run id */  
    p_event_builder in number, /* number of the event builder */  
    p_events      in number, /* number of events */  
    p_bytes       in number, /* file size in bytes */  
    p_file_stop   in date ); /* stop time of the file */
```

The procedure first updates the not existing data (number of events, file size and stop time) in the table daq_run. It then checks if the stop time exists already in the table DAQ_RUN_COMMON or if the existing stop time is larger than the new stop time and at least equal to the common start time.. Eventually the data are changed. At the end, after all PERL scripts inserted their data for a run, the common stop time is the minimum stop time of all files generated by the different event builders.

Again, any error (file does not exist, file stop exists already) is recorded in the table DAQ_RUN_ERROR_LOG. The “normal” update is discarded.

The program does not allow a user other than the owner DAQ to change the stop time of a file, which already has a stop time in the table. This guarantees, that old data cannot be spoiled by mistake.

2.3 WebDB interface for DAQ files

The new application “**Show DAQ files**” in folder “**DAQ and Beamtime**” shows the list of DAQ files in a selectable time interval. It is possible to set constraints on the run type, the minimum number of events, the minimum magnet current and/or a filename substring.

2.4 WebDB interface for the DAQ operator

The new interface “**Show event builders**” in folder “**DAQ and Beamtime**” allows the DAQ operator

- to see the data of all files,
- to check the consistency of the filename and the run id,
- to check the time differences between the different event builder start and stop times (A large difference indicates, that the local clocks of the event builders are not properly synchronized.),
- and to check the error log.

3 New and changed user interfaces

Based on the new tables several new views and utility functions were developed and existing views and code were replaced. But in this document only the new and changed user interfaces with new functionality will be described in more detail.

3.1 Beam time logbook

The code for the beam time logbook was rewritten. The layout of the form was changed and a field added for text search. It allows to search simultaneously for multiple text pattern (eventually with logical expressions) independent of the order. Highlighting might be implemented in the future using a JAVASCRIPT library.

The logbook shows the DAQ runs with the common start/stop times, the common substring of the filenames, the number of files and the sum of the events, but not the complete list of individual files for the same run id.

A link allows now to reload the form and to actualize the drop-down menus keeping all user settings.

3.2 Statistic plots for DAQ files

The new application “**Charts of beam time and run statistics**” in folder “**DAQ and Beamtime**” shows for a selected beam time and run type scaled bar charts (for a whole beam time or a selected day) of

- the number of runs,
- the number of files,
- the number of events,
- the number of event builders,
- the measuring time,
- the mean event rate.

Different colors indicate the mean magnet current. An additional table shows the summary information in the selected time interval.

3.3 Run validation

The actual concept is to validate not individual DAQ files, but DAQ runs (all files with the same run id).

The WebDB interfaces show typically the runs, the number of files, the common substring of the filenames and the sum of the events in all files. Only the file list for the analysis (in ASCII format) shows the complete list of files.

If a lower limit on the number of events is specified, it puts a constraint on the sum of the events, not on the number of events in a single file.

Remark: If it turns out in the future, that we need to validate individual files, the necessary changes could be implemented within 1 - 2 weeks.

Not yet implemented:

J. Pietraszko proposed to add an additional “detector” DST to store also the runs used for the DST

production independently of the quality of the individual detectors, because a run marked as restricted for a detector might show no major change in the QA plots and can be therefore used in the DST production.

3.4 Slow control data

Slow control summaries:

Slow control summaries are not generated for individual DAQ files, but DAQ runs (all files with the same run id).

The time range corresponds to the minimum start and stop times of the different event builders. All programs and WebDB interfaces were changed to work with the new DAQ tables.

Storage of meta data for rarely recorded channels:

Triggered by RPC for its way to modify thresholds, a new set of tables, programs and interfaces was developed to extract once per day the slow control data of special channels and to store them in a separate table. These data do not have a single time stamp, but a time range for fast access. The interfaces read the data valid at the run start.

This new type of storage could be used in the future not only for RPC thresholds, but for all other interesting channels (as for example beam parameters), which change only a few times during a beam time and only the data at the begin of a run are relevant.

3.5 Log of HADES magnet and cryo plant

All existing WebDB applications in folder “Magnet and Cryo” were replaced by new applications in the Oracle production account CRYO.

The old packages were once generated with the WebDB toolkit on the development account KOENIG (private account by Wolfgang Koenig).

Remark: It turned out, that the WebDB toolkit for charts contains several bugs.

The layout of the new interface has changed, but not the functionality.

The WebDB folder was reorganized for easy distinction between the magnet and the cryo log, because the data of the cryo plant are not stored anymore in Oracle since Spring 2004.

3.6 Simulation

The run id generator used in the past by the DAQ and in the HADES simulation package HGEANT was originally introduced to have the possibility to store also simulation runs in Oracle and to analyze the data without the need to specify explicitly the reference run belonging to each project. Although interfaces existed since many years, simulation runs, event generator files and the configuration files for the simulation and the event generators were never stored in Oracle.

The fact, that in the future the DAQ will generate the run ids from the time stamp, does not allow anymore to use the run id generator in the simulation package. The storage of simulation runs is therefore not possible anymore (at least not with major modifications). The interfaces have been removed meanwhile from the analysis (library orautil) and also the WebDB interfaces were changed.

In HGEANT the run id generator was discarded and in the future all simulation runs will store directly the reference run in the event header. This also allows an automatic initialization in the analysis.

3.7 Analysis interface

All changes in Oracle were done such, that old HYDRA code compiles and works properly without any changes, not only for runs taken in the past, but also for future runs with one or several event builders.

The initialization parameters for all files with the same run id, but created by different event builders, are identical.

Complete filenames, which include also the event builder number, are accepted, but only the first 13 characters (prefix + time string) are relevant. Some interfaces, which show besides the run id also the file name (for example the log file created by the parameter file generator or the slow control info) show now only this common substring of the files and the sum of all events taken by the different event builders.

If needed, a new function might be added in class HOraInfo to read the complete list of files.

But one should have in mind, that run validation is not done for individual files, but for runs (all files with the same run id). Also the slow control data do not distinguish between the different event builder files. **Therefore eventually some macros need to be modified.**

3.8 Interface to show and change experiments

The old interfaces made once with the WebDB toolkit were replaced by new applications.

The public interface “**Experiment Info**“ in WebDB folder “**DAQ & Beamtime**” shows the name, location, begin, end, purpose and description of beam times and/or simulation projects.

Experiments can be added or changed with the new application “Show, insert and change experiments” in sub-folder “Beam time maintenance”. But the access is restricted to special Oracle accounts.