



# Run Quality Bookkeeping

*RAL Computing Workshop 01-13-2003*

Alessio Sarti

INFN and University of Ferrara

# RQM work

The RQM work in BaBar can be summarized as follows:

“Starting from:

- list of swept collections & BOOT federation
- quality information for each run

the RQM must provide the physicists in the collaboration a reliable list of runs *good* for analysis.” The full chain works that way:

- Run is taken in IR2. *DQM shifter sets a quality flag*
- Run is shipped & processed to/by PC/ER farms
- .....

# RQM work (cont.)

- .....
- *Data collections are swept* from production federation to the analysis federations
- QA jobs (testing database problems) are submitted against the swept collections
- A run list is produced for those runs that have a *'good' quality flag* and that *survived QA jobs*
- The good runs list is made accessible to users: they can use it to constrain queries on data samples

# RQM Bookkeeping

## Where the bookkeeping enters in RQM work?

- Storing run's quality information *in a way easy to be used in data sample queries*
- Tracking 'good runs' lists and datasets: what are the problematic runs, why did they crashed, etc. etc.
- Keeping track of QA jobs output as a function of different variables: sweep week, farm, release (existing tool that checks/resubmit crashed jobs)
- The current setup is **not dealing easily with recovery of bad runs**: recreate consistent good runs lists is a painful work (oracle usage can help).

# User bookkeeping

As RQM (running QA jobs) and member of ISL group (running ISL jobs) I've gained also a bit of 'user' experience. Main needs are:

- Have an easy way of constraining queries to ONLY runs marked *good* for analysis (DATA)
- Have a flexible tool that handles complex queries with a **well documented syntax** (both DATA and MC)
- Track datasets in time and (much more interesting from the conference point of view :)... have an easy way of accessing 'frozen' data samples.

# Run's quality informations

How to provide to the users the quality information for each run?

- Tools that are creating runs/collections lists can access that information via Oracle/mySql databases (now they're using lists in .txt format)

How to track changes of information vs time?

- RQM based: use cvs or Oracle/mySql to store 'frozen' lists?
- User based: creation of 'log' files? Use a tool that get the difference btw various *centralized* 'snapshots'?

# Tracking datasets

Data sample changes (at last) once a week: how to deal with different 'tracking' requests?

- Few people that needs a continuous update on available data (ex.: OPR code testing, detector problems divesting, etc. etc). *Tools for run list creation are good enough for handling that provided that they're synchronized with sweeps*
- The bulk of the collaboration (mainly AWGs) that needs updates on *consistent data sample changes* and works mainly with 'frozen' data samples. A repository with run lists easily accessible by user's tools is needed.

# Queries

Users that want to access data are using queries. Main query fields usually are:

- Quality flags, detector conditions, processing releases, run ranges, collection's number of events, official datasamples....
- Different levels of complexity can be observed: mostly the user wants just tcl files for *good runs of on/off resonance skimmed data approved for use in the current conference cycle, processed with the latest official software releases*, but we need to provide easy access to all the other informations

# Status: RQM

How are we currently dealing with those needs?

- Each run (once available in analysis federations) is documented in a 'run / procspec' list (.txt file) and the corresponding 'AllEvents' collection tcl file is created (.tcl file). Lists are stored in cvs (GoodRuns pkg).
- Data sample & run lists are sampled by the RQM in blocks/subsets accordingly to detector/releases changes: documentation is in dataset web page.
- A FAQ page is provided in data sample page: contains documentation of some common issues encountered while dealing with data/MC sampling and queries

# User: lists creation tools

From user's point of view there are two main concern: be aware of data sample changes & have an easy way to create run/collection lists.

- Currently available tools are handling various queries: run range, release, onpeak off peak, goodruns (*Attention: now the goodruns constraint is weaker than the runrange one*).
- MC queries needs also to be considered: most of the 'common requirements' are similar but there are specific issues (skim naming, MC conditions, etc. etc.).
- **Difficult to match detector and MC conditions**

# User: tracking datasample

How to integrate datasample information tracking & creation of collections lists?

- Web/cvs documentation tracks datasamples: conditions, luminosities, run ranges,... Easy to browse & cvs tag related
- There's no automated tool that can be used in lists creation that track down to run number the conditions for each run. No easy tool is available creating tcl files for a given 'detector condition'. (getGoodData.pl is now deprecated)

# Documentation

Documentation plays an important role in that game & can be improved:

- Datasets must be well documented. Information must be provided in a way that can be also used as input for user's queries: enlarge usage of Oracle/mySql databases.
- Tools must be documented with a large number of examples, a FAQ page and a complete and useful help.
- BaBar data page and skimData page are two 'useful' starting point to get informations on how-to deal with data/MC samples.

# The perfect tool

Today's bookkeeping tools are still missing some 'useful' features. A better tool would be the one that:

- knows by itself the list of good runs using quality information stored in oracle (we get rid of collections lists stored in cvs).
- knows by itself the list of swept collections (again: oracle?).
- can handle a request on number of events / collection (needed to avoid jobs crashed for CPU time reasons)
- knows about detector conditions: sampling can be made by RQM and persisted (again: oracle?)

# What we have done so far

With respect to Run2 the tools/documentation status is improved:

- Full chain from IR2 to the user has been revised and frozen for Run3 data: this implies a major responsabilization of DQM shifters.
- The information about datasets is stored in cvs and accessible via web: this makes more easy datasample changes tracking.
- A powerful tool (skimData) is available for data and MC collection lists creation: that tool has a good documentation and can handle sophisticated queries

# What is still missing

Tools used to handle bookkeeping can still be improved. On the query side we need to add the capability to query the datasample for:

- detector conditions
- number of events in collections
- swept collections

Bookkeeping tools used by RQM must also handle sanity checks on swept collections:

- database corruptions
- duplicated events

# Conclusions

- In the future RQM role should be focused in setting the 'right' quality flag in oracle, instead of releasing tcl files.
- Automation of bookkeeping tools performing sanity checks on swept collections can help RQM work
- Documentation of data samples is another crucial task: the best way that can be used to store the information must be discussed.
- Keeping track of data sample changes using cvs is useful from RQM point of view: need to be understood if it's true also for users...