Table of Contents

THIS PAGE NEEDS REVISION.

Frequently Asked Questions about the Conditions Database

This FAQ presents answers to questions regarding the functionnalities and usage of the Conditions Database and the Condition management framework of LHCb. We hope it to be as up to date as possible, as the CondDB project is presently in a strong mutation phase. Do not hesitate to contact me if your question is not answered below.

Some Definitions

- **What is a condition ?**
  We call *condition* any parameter (or set of parameters) used for data reconstruction or analysis which is varying in time in a non regular way. A condition is thus a serie of values valid for a given period of time: the *interval of validity*.
  Examples of conditions: alignment constants, atmospheric pressure, trigger configuration, etc.

- **What is the Conditions Database ?**
  Well, it is a database containing conditions ;-). It will basically store *all* and *only* the conditions which are needed by LHCb reconstruction and analysis software. The conditions values stored are accessible with three "co-ordinates": the *data item*, i.e. the name of the condition, the *version* and the *time* (cf. figure below).

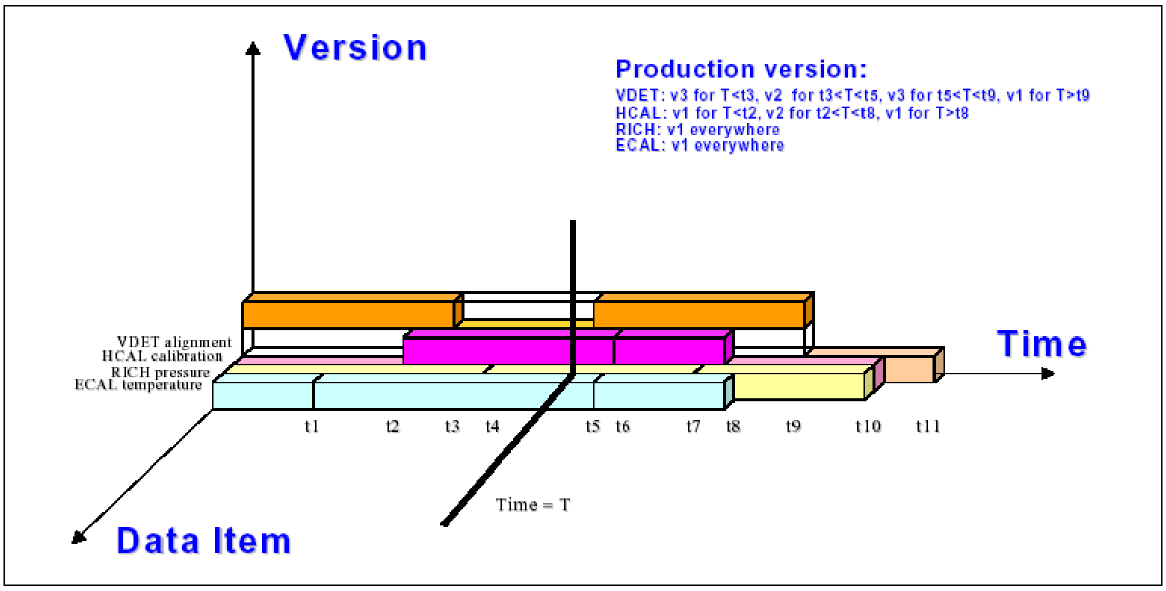

Figure 1 The three axes for identifying uniquely each data item in the condition database

- **What do you mean by "online" and "offline" ?**
  When refering to *online*, we refer to everything that is synchronised with data taking. The Experiment Control System and the triggers processes are typical examples of online activities. All the rest is considered as *offline*. Full reconstruction and data analyses are offline activities.

Condition Storage

- **Where do the conditions come from ?**
  With the exception of VELO alignment constants and online algorithms configuration, the conditions

produced online are exclusively produced by the Detector Control System, i.e. *hardware*. Offline, the conditions will be produced exclusively by *software*, e.g. reconstruction algorithms.

- **How is the Conditions Database populated ?**
  The Experiment Control System is responsible to format and write the conditions produced online in the CondDB. The conditions produced offline will be stored in the CondDB via a Gaudi service.

Accessing to the Conditions Database

- **How do I access the Conditions Database ?**
  Only some very specific Gaudi services and converters will access the CondDB directly. Algorithms actually won't know about the existence of the CondDB itself. They will simply ask GAUDI services for up to date conditions and don't have to bother where they come from.

- **What is the Master Condition Database ?**
  It is the official CondDB used for offline applications. It will contain all the conditions produced online and the conditions produced offline which have been approved by super users. All the CondDB distributed on the GRID will be total or partial copies of the Master CondDB.

- **Who is allowed to write in the Master Conditions Database ?**
  Conditions produced online will be stored in a special instance of the CondDB wich will be automatically synchronised with the Master CondDB. Appart from this "automatic" writting process, a small number of super users will have the right to edit the CondDB, either to put new approved conditions values, or set new tags, etc. It is unlikely that *removing* data from the CondDB will be allowed.

- **What about the "standard users" ? Where can they store new conditions ?**
  Standard users will have the possibility to create their own instance of the Master CondDB, either total or partial, and be able to manage it as they wish. This means that they can write their own conditions in it, apply tags, etc. whithout risk of causing trouble to the Master CondDB

- **How do a user will broadcast new conditions ?**
  If a user thinks that the conditions he has produced might be of interest for the community, then he will submit them to a super user for approval. If approved, the super user will update the Master CondDB.
  The submission protocol is still to be defined.

- **Is it possible to read the Conditions Database ?**
  A graphical interface will be available to browse a CondDB. However, *editing* the CondDB will be strongly restricted in order to protect data integrity.

Using the Conditions

- **Where do I find examples?** In the LHCb package Ex/DetCondExample. A good idea is to start from the two files ExampleAlg.h and ExampleAlg.cpp. They are a good example of how one can use conditions and the Update Manager Service from a GaudiAlgorithm (or a GaudiTool).

- **How do I use the conditions?**
  Actually only the users can answer to this questions. When you ask to the detector data store a condition, you will get a pointer to a `Condition` object which is essentially a `ParamValidDataObject`. You can access the informations inside a `Condition` object using the `paramAs*` methods.
  An example (from an algorithm):

```
Condition* myCond = getDet<Condition>("/dd/Conditions/Environment/MyDet/Crate1");
```

```
double crate1Temp = myCond->param<double>("Crate1Temperature");
```

- **How to register for a condition to the Update Manager Service?**
  The `UpdateManagerSvc` is a service (of course) that *notifies* objects of changes in conditions.
  An object (Algorithm, Tool... anything) has only to tell the `UpdateManagerSvc` which method to call when which condition changes. Each object may have many methods that have to be called when conditions change and each mathod can depend on many conditions. It is also possible to tell the `UpdateManagerSvc` that you want a condition to be always up-to-date, but you do not need a method to be called.
  In some cases, it can be useful to be notified when an object, which depends on a condition, has been updated. For those cases it's enough that the object has already been register to the `UpdateManagerSvc`.
  An example is in ExampleAlg.h and ExampleAlg.cpp.

- **How do I make a condition value available to other running algorithms ?**
  Conditions are always available to all agorithms since they are in the detector data store. If you want to create your own Condition objects and allow other algorithms to access them, you just have to register the new conditions to the detector data service.
  If you want to change the numbers inside a Condition object and you want to trigger an update of all the objects depending on that condition, you can use the `invalidate` method of `UpdateManageSvc`.
  Usual example:

```
Condition* myCond = getDet<Condition>("/dd/Conditions/Environment/MyDet/Crate1");
myCond->addParam<double>("Pressure1",1024,"comment");

updMgrSvc()->invalidate(myCond);

// if you don't want to wait for the next event before the update...
updMgrSvc()->newEvent();
```

-- NicolasGilardi - 28 Apr 2005

-- MarcoClemencic - 03 May 2005

-- MarcoClemencic - 09 Feb 2006

This topic: LHCb > FAQCondDB
Topic revision: r11 - 2013-03-20 - IllyaShapoval