Table of Contents

Recommended

Introduction

This is based on Thomas Ruf's talk at T-rec on 13/7/09 ⬚.

Which database tags should I use?

As a general rule, for real data it should always be safe to use in the analysis the latest available LHCBCOND, DDDB and DQFLAGS global tags. The latter has a direct effect on which events are processed. So it is recommended to use the latest to use the best knowledge of the Data Quality. Unfortunately this may not always be possible due to incompatibilities in the software that we try to avoid but which sometimes creep in nevertheless. For simulation, you should use the same DDDB and SIMCOND tags as in the production, with the same caveat about software compatibilty.

Why different tags for MC?

For real data, new tags mean better description of existing detector. For MC data, the best description is the one used during the simulation step. Therefore, why different Tags for MC ?

- Special settings, Velo open, B field off, ... are identified with a Tag and this Tag needs to be used when reconstructing and analysing the data
- There can be changes in the software which require changes in the databases. Newer software versions might not work anymore with old Tag of database. Requires some intelligence for defining compatible tags.

For MC it is useful to employ the tags that where used to produce the simulation data files:

for this one can use the book keeping information provided by Fecim the LHCb Bookkeeping Browser: for a given file

one can just cick (mouse right button) to get the "More information" to retrieve the conDB and DDDb tags

Why is setting tags required?

Which database tag to use may change, such as when a new mandatory datapoint is added, or when a genuine and post-correctable error is noticed, or if there are updates on ancilliary information, such as (in the future) luminosity information.

So in general it is best to allow the tags to change, which is difficult in the book-keeping, so requires user-input. However, core-software is working on a solution to this issue where the tags are stored in the file itself, and may be overwritten with a centrally maintained map should such updates occur.

Knobs

DataType, DDDBtag, CondDBtag, Simulation flag. The DaVinci and Brunel configurable has a variable called DataType, which is the main knob for most use cases. The information is forwarded to the LHCbApp configurable. Known DataTypes are:

```
LHCbApp().DataType = 'MC09' # or 'DC06, '2008', '2009', ..., '2018'
```

If nothing else specified, the LHCbApp will be instrumented to use the default tag:

```
LHCbApp().DDDBtag = 'default'
LHCbApp().CondDBtag = 'default'
```

The simulation flag is used to switch from LHCbCONDto SIMCOND database for conditions.

```
LHCbApp().Simulation = 'True' # or 'False'
```

In some cases, 'DC06', 'MC09', implicitly set by choice of DataType.

To get the latest tags according to DataType for every partition, one can specify

```
from Configurables import CondDB
CondDB( LatestGlobalTagByDataType = "2016" )
```

To get the latest tag according to DataType for selected partitions (available for versions released after September 2016) , one can specify

```
CondDB( Tags = {"LHCBCOND":"<latest:2016>",
                "DDDB":"<latest:2016>",
                "DQFLAGS":"<latest:2016>"} )
```

E.g. one could set production specific tags for CondDB and DDDB as specified above and use the latest Data Quality tags.

Leaving the choice of Tag to the DDDB configurable will solve most use cases. Only needed to specify the DataType corresponding to the data to be used. NOTE: Only one set of tags can be used in one execution. It is not possible to mix data which would require different Tags.

Default tags

The default tag is resolved by DDDB/Configuration.py. Which means, the actual Tag used depends on the version of LHCb. The person changing DDDB/Configuration.py has to make sure that a new Tag is compatible with the DataType. For example LHCB v27r3:

- 'DC06': "DC06-20081002" for DDDB and LHCBCONDNote: SIMCOND does not exist for DC06, Simulation flag set to False, bfield polarity = -1 forced
- '2008': "head-20090330" for DDDB, "head-20090402" for LHCBCOND and "sim-20090212" for SIMCOND
- 'MC09': "MC09-20090602" for DDDB and "sim-20090402-vc-md100" for SIMCOND, Simulation flag ignored
- '2009': "head-20090617" for DDDB, "head-20090508" for LHCBCOND and "sim-20090508-vc-md100" for SIMCOND. Simulation flag for switching between SIMCOND and LHCBCOND

Where do I find information about used tags:

- Start of job, tells what will be used:
  - ♦ # WARNING: Default tag requested for partition DDDB (using DC06-20081002)
  - ♦ # WARNING: Default tag requested for partition LHCBCOND (using DC06-20081002)
- In each event, Rec/Header for example contains information which Tag had been used during reconstruction
  - ♦

```
>>> evt['Rec/Header']
```

- - - ◊ { randomSeeds : [340212, 5501, 540002744, 0] ...
    - ◊ condDBTags : [(DDDB, head-20081002), (SIMCOND, head-20081002)]
- Bookkeeping using dirac tools (SetupProject Dirac, lhcb-proxy-init):
  - dirac-bookkeeping-production-information 4918
    - ◊ Step0:Gauss-v37r2 Option files: $APPCONFIGOPTS/Gauss/MC09-b5TeV-md100.py;$APPCONFIGOPTS/Conditions/MC09 **DDDb: MC09-20090602 CondDb: sim-20090402-vc-md100**; ...
    - ◊ Step2:Brunel-v34r7 Option files: $APPCONFIGOPTS/Brunel/MC09-WithTruth.py; $APPCONFIGOPTS/Conditions/MC09-20090602-vc-md100.py **DDDb: MC09-20090602, CondDb: sim-20090402-vc-md100**
- Bookkeeping GUI (lhcb_bkk)
  - If you know the production ID, check the "Production Lookup" check box, type in the production ID and click OK. This gives a table with the same information as the dirac-bookkeeping-production-informations tool.
  - Otherwise, locate a processing pass, e.g. LHCb - Collision09 - Beam450GeV-VeloOpen-MagDown - RealData+RecoToDST-07+2009-Stripping02
    - ◊ Right click on the processing pass name and choose "More Information"
    - ◊ Click on the tab corrsponding to the step you are interested in (e.g. 2009-Stripping02 in above example)
- DIRAC monitoring
  - Go to the dirac monitoring⊠, look for the production ID (it's part of the LFN), click on the production and ask for "Show details". Voilà
- Bender
  - Use the command "lb-run Bender/latest db-tags FILE", where FILE can be the LFN or PFN. It prints you the database tags used in production.
- Production twiki
  - A list of all productions including database tags is maintained here

Where do I find information about global tags

- All available tags are listed in the CondDB release notes, http://lhcbdoc.web.cern.ch/lhcbdoc/dbase/conddb/release_notes.html⊠
- Example 1 "DC06-20081002":
  - DDDB: based on "DC06-20080930"
  - LHCBCOND: based on "DC06-20080930" plus:calo-DC06-20081002 …
    - ◊ Global tag "DC06-20080930": DDDB: based on "DC06-20080904" plus:calo-20080923, calo-20080905
      - · 2008-09-23 - Olivier Deschamps
      - · DDDB: calo-20080923 (show files)
      - · LHCBCOND: calo-20080923 (show files)
      - · SIMCOND: calo-20080923 (show files)
      - · Reorganized conditions with some new ones.
- Example 2 "MC09-20090602":
  - DDDB: based on "MC09-20090526" plus: ot-20090519
    - ◊ DDDB: ot-20090519 Fixed wrong material for the Roha component of a coupling.
      - · Global tag "MC09-20090526": DDDB: based on "MC09-20090519" plus:velo-20090526, particles-20090526

When should I use the Oracle CondDB?

NEVER! In the past, it was necessary to use Oracle to get the most recent Online conditions when analysing recent real data. These conditions are no longer distributed via Oracle: now, up to date snapshots are

automatically made available at the end of each run as part of the standard distribution of SQLite files via the cvmfs file system.

-- ThomasRuf - 14 Jul 2009 -- MarcoCattaneo - 29-Oct-2012 -- VincentTisserand - 15-Mar-2011

This topic: LHCb > RecommendedTags
Topic revision: r21 - 2019-01-22 - AdamMorris