Introduction to Conditions DB & Update Manager

Marco Clemencic

marco.clemencic@cern.ch
Overview

- Conditions Database
  - Key concepts
  - Definitions
  - Organization
- Update Manager Service
  - What it is for
  - Main features
Conditions Database
Key concepts

Tool to handle conditions

- Conditions
- IOVs
- Folders

- Single-Version
- Multi-Version
Definitions

- **Condition**: a datum which is valid for a defined period
- **IOV**: Interval Of Validity, the period for which a condition is valid
- **Folder**: container of condition with the same meaning
- **Single-Version Folder**: Folder that can contain only non overlapping IOVs
- **Multi-Version Folder**: Folder that can contain overlapping IOVs (a set of IOVs is identified by a TAG)
The conditions database is organized like a filesystem.

Folders are the equivalent of a file

A folder is identified by its path

/Conditions/LHCb/Environment/Condition1
/Conditions/LHCb/Environment/Condition2
/Conditions/Velo/Alignment/Station1
/Conditions/Velo/Alignment/Station2
Update Manager Service
Data in the CondDB are mapped to objects in the Transient Detector Store

We need to keep the TDS synchronized with the event time:
- we do not want to scan the whole TDS every event to find which conditions need an update
- users may want to react to changes with provided code

The Update Manager Service is the service performing the updates and calling user code when needed.
Main features

- Objects needing a conditions (consumers) must be registered to the UMS and optionally provide a call-back function
- “Conditions” can be anything inside the TDS or anything known by the UMS (e.g. another consumer)
Conclusions
Conclusions

- Normal users do not interact with the CondDB
- they just need to talk to the UMS
- The UMS is very flexible and easy to use
  … at least I hope so 😊

Let’s move to the tutorial!