

Using the COOL Conditions Database in the cosmics

Refer also to:

- SCT_ConditionsData
- SCT_ConditionsAlgs

Features

Masking of Missing Modules in SCT_Digitization

The proposed cosmic setup uses 504 barrel modules arranged in two phi segments. Up to and including version 00-00-19 of SCT_ConditionsAlgs, the tool SCT_ModuleStatusAlg was hard coded with the offline locations of these modules. SCT_Digitization would then retrieve the SCT_ModuleStatusAlg tool and thus determine if a given hit (real or noise) took place in a module which was present in the setup.

The new version of SCT_ModuleStatusAlg, found in SCT_ConditionsAlgs-00-01-01, fills a map of the barrel modules which are used from a COOL table. The table has 504 channels and as its payload holds the module serial numbers and their offline identifiers. After the map is filled (through a callback) it can be interrogated for each hit which is processed in SCT_Digitization.

Accessing Module serial numbers

If the map of the modules used in the cosmic arrangement is filled from the database, the mapping of module serial numbers to offline positions is known. With the module serial numbers available in the offline, it will then be possible in the future to access further conditions data associated with the modules.

Alternatively, if the database is not used, the offline locations of the modules present are known but not their serial numbers.

Usage

The methods

To use the tool, for example in SCT_Digitization, the SCT_ModuleStatusAlg tool must first be retrieved. The method getModuleSn(vector) can then be called, which given the identifiers layer, phi, eta (barrel only for now) returns the module serial number if present. If the map was not filled from the DB, the serial number is returned as "UNKNOWN" if the module is present.

Controlling job options

In `InDetCosmicDig_topOptions.py` the user chooses whether the module mapping is filled from the database or not with the flag:

```
SCTUseCondDb = True
```

The arrangement is also chosen:

```
SCTCosmicSetup = 2
```

NOTE: The choice of arrangement only applies when **not** using the database. The modules in the database assume arrangement type 2.

The above arguments are passed to `InDetCosmicDig_jobOptions.py` which in turn set flags in the `SCT_ModuleStatusAlg` tool. From the point of view of `SCT_Digitization`, it does not matter whether the DB is used or not:

```
ToolSvc.SCT_ModuleStatusAlg.CosmicSetup = SCTCosmicSetup
ToolSvc.SCT_ModuleStatusAlg.ModuleCondDb = SCTUseCondDb
if ToolSvc.SCT_ModuleStatusAlg.ModuleCondDb:
    include("InDetCosmicSimExample/InDetCosmicDig_condDbOptions.py")
    include("SCT_ConditionsAlgs/SCT_ConditionsReadMapping.py")
```

The job option fragment `InDetCosmicDig_condDbOptions.py` specifies the database connection.

- `InDetCosmicDig_condDbOptions.py`
- `InDetCosmicDig_jobOptions.py`
- `InDetCosmicDig_condDbOptions.py`

-- Paul Bell - 04 Sep 2005

This topic: `Main > CosmicCondDb`

Topic revision: `r4 - 2005-09-06 - Paul Bell`



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback