

From CMSSW/DataFormats/DetId/interface/DetId.h

```

001 #ifndef DATAFORMATS_DETID_H
002 #define DATAFORMATS_DETID_H
003
004
005 //FIXME shall be removed and implemented where the operator is defined
006 #include <ostream>
007
008 #include <stdint.h>
009 /** \class DetId
010
011 Parent class for all detector ids in CMS. The DetId is a 32-bit
012 unsigned integer. The four most significant bits ([31:28]) identify
013 the large-scale detector (e.g. Tracker or Ecal) while the next three
014 bits ([27:25]) identify a part of the detector (such as HcalBarrel
015 (HB) for Hcal).
016
017 $Date: 2010/08/23 12:56:41 $
018 $Revision: 1.1 $
019 */
020 class DetId {
021 public:
022     static const int kDetOffset      = 28;
023     static const int kSubdetOffset   = 25;
024
025
026     enum Detector { Tracker=1,Muon=2,Ecal=3,Hcal=4,Calo=5 };
027     // Create an empty or null id (also for persistence)
028     DetId() : id_(0) { }
029     // Create an id from a raw number
030     DetId(uint32_t id) : id_(id) { }
031     // Create an id, filling the detector and subdetector fields as specified
032     DetId(Detector det, int subdet) {
033         id_=((det&0xF)<<28)|((subdet&0x7)<<25);
034     }
035
036     // get the detector field from this detid
037     Detector det() const { return Detector((id_>>kDetOffset)&0xF); }
038     // get the contents of the subdetector field (not cast into any detector's numbering enum)
039     int subdetId() const { return ((id_>>kSubdetOffset)&0x7); }
040
041     uint32_t operator()() const { return id_; }
042     operator uint32_t() const { return id_; }
043
044     // get the raw id
045     uint32_t rawId() const { return id_; }
046     // is this a null id ?
047     bool null() const { return id_==0; }
048
049     // equality
050     bool operator==(DetId id) const { return id_==id.id_; }
051     // inequality
052     bool operator!=(DetId id) const { return id_!=id.id_; }
053     // comparison
054     bool operator<(DetId id) const { return id_<id.id_; }
055
056 protected:
057     uint32_t id_;
058 };
059
060 // equality
061 inline bool operator==(uint32_t i, DetId id) { return i==id(); }
062 inline bool operator==(DetId id, uint32_t i) { return i==id(); }
063 // inequality
064 inline bool operator!=(uint32_t i, DetId id) { return i!=id(); }

```

DetId < Main < TWiki

```
065 inline bool operator!=(DetId id, uint32_t i) { return i!=id(); }
066 /// comparison
067 inline bool operator<(uint32_t i, DetId id) { return i<id(); }
068 inline bool operator<(DetId id, uint32_t i) { return id()<i; }
069
070
071 //std::ostream& operator<<(std::ostream& s, const DetId& id);
072
073 #endif
```

-- DavidCockerill - 23-Aug-2010

This topic: Main > DetId

Topic revision: r1 - 2010-08-23 - DavidCockerill



Copyright &© 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback