CMSSW/ DataFormats/ EcalDetId/ src/ EEDetId.cc

```
001 #include "DataFormats/EcalDetId/interface/EEDetId.h"
002 #include "FWCore/Utilities/interface/Exception.h"
003
004 #include <iostream>
005 #include <algorithm>
006
007 const int EEDetId::QuadColLimits[EEDetId::nCols+1] = { 0, 8,17,27,36,45,54,62,70,76,79 };
008
009 const int EEDetId::iYoffset[EEDetId::nCols+1]      = { 0, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0 };
010
011 const unsigned short EEDetId::kxf[] = {
012   41,  51,  41,  51,  41,  51,  36,  51,  36,  51,
013   26,  51,  26,  51,  26,  51,  21,  51,  21,  51,
014   21,  51,  21,  51,  21,  51,  16,  51,  16,  51,
015   14,  51,  14,  51,  14,  51,  14,  51,  14,  51,
016    9,  51,   9,  51,   9,  51,   9,  51,   9,  51,
017    6,  51,   6,  51,   6,  51,   6,  51,   6,  51,
018    6,  51,   6,  51,   6,  51,   6,  51,   6,  51,
019    4,  51,   4,  51,   4,  51,   4,  51,   4,  56,
020    1,  58,   1,  59,   1,  60,   1,  61,   1,  61,
021    1,  62,   1,  62,   1,  62,   1,  62,   1,  62,
022    1,  62,   1,  62,   1,  62,   1,  62,   1,  62,
023    1,  61,   1,  61,   1,  60,   1,  59,   1,  58,
024    4,  56,   4,  51,   4,  51,   4,  51,   4,  51,
025    6,  51,   6,  51,   6,  51,   6,  51,   6,  51,
026    6,  51,   6,  51,   6,  51,   6,  51,   6,  51,
027    9,  51,   9,  51,   9,  51,   9,  51,   9,  51,
028   14,  51,  14,  51,  14,  51,  14,  51,  14,  51,
029   16,  51,  16,  51,  21,  51,  21,  51,  21,  51,
030   21,  51,  21,  51,  26,  51,  26,  51,  26,  51,
031   36,  51,  36,  51,  41,  51,  41,  51,  41,  51
032 } ;
033
034 const unsigned short EEDetId::kdi[] = {
035     0,   10,   20,   30,   40,   50,   60,   75,   90,  105,
036   120,  145,  170,  195,  220,  245,  270,  300,  330,  360,
037   390,  420,  450,  480,  510,  540,  570,  605,  640,  675,
038   710,  747,  784,  821,  858,  895,  932,  969, 1006, 1043,
039  1080, 1122, 1164, 1206, 1248, 1290, 1332, 1374, 1416, 1458,
040  1500, 1545, 1590, 1635, 1680, 1725, 1770, 1815, 1860, 1905,
041  1950, 1995, 2040, 2085, 2130, 2175, 2220, 2265, 2310, 2355,
042  2400, 2447, 2494, 2541, 2588, 2635, 2682, 2729, 2776, 2818,
043  2860, 2903, 2946, 2988, 3030, 3071, 3112, 3152, 3192, 3232,
044  3272, 3311, 3350, 3389, 3428, 3467, 3506, 3545, 3584, 3623,
045  3662, 3701, 3740, 3779, 3818, 3857, 3896, 3935, 3974, 4013,
046  4052, 4092, 4132, 4172, 4212, 4253, 4294, 4336, 4378, 4421,
047  4464, 4506, 4548, 4595, 4642, 4689, 4736, 4783, 4830, 4877,
048  4924, 4969, 5014, 5059, 5104, 5149, 5194, 5239, 5284, 5329,
049  5374, 5419, 5464, 5509, 5554, 5599, 5644, 5689, 5734, 5779,
050  5824, 5866, 5908, 5950, 5992, 6034, 6076, 6118, 6160, 6202,
051  6244, 6281, 6318, 6355, 6392, 6429, 6466, 6503, 6540, 6577,
052  6614, 6649, 6684, 6719, 6754, 6784, 6814, 6844, 6874, 6904,
053  6934, 6964, 6994, 7024, 7054, 7079, 7104, 7129, 7154, 7179,
054  7204, 7219, 7234, 7249, 7264, 7274, 7284, 7294, 7304, 7314
055 } ;
056
057 EEDetId::EEDetId( int index1, int index2, int iz, int mode ) : DetId( Ecal, EcalEndcap )
058 {
059    int crystal_ix=0;
060    int crystal_iy=0;
061    if (mode == XYMODE)
062    {
063       crystal_ix = index1;
064       crystal_iy = index2;
```

```
065    }
066    else if (mode == SCCRYSTALMODE)
067    {
068       int SC = index1;
069       int crystal = index2;
070       //      std::cout << "iz " << iz << " SC " << index1 << "crystal " << index2  << std::e
071
072       crystal_ix=iz*ix(SC,crystal);
073       if (crystal_ix<0)
074         crystal_ix++;
075       crystal_ix+=50;
076       crystal_iy=iy(SC,crystal);
077       if (crystal_iy<0)
078          crystal_iy++;
079       crystal_iy+=50;
080
081    }
082    else
083    {
084       throw cms::Exception("InvalidDetId") << "EEDetId:  Cannot create object.  Unknown mode
085    }
086
087    if (!validDetId(crystal_ix,crystal_iy,iz))
088    {
089       throw cms::Exception("InvalidDetId") << "EEDetId:  Cannot create object.  Indexes out o
090                                            << "x = " << crystal_ix << " y = " << crystal_iy <
091    }
092
093    id_|=(crystal_iy&0x7f)|((crystal_ix&0x7f)<<7)|((iz>0)?(0x4000):(0));
094 }
095
096 EEDetId::EEDetId( const DetId& gen )
097 {
098    if (!gen.null() && (gen.det()!=Ecal || gen.subdetId()!=EcalEndcap)) {
099       throw cms::Exception("InvalidDetId");
100    }
101    id_ = gen.rawId();
102 }
103
104 EEDetId& EEDetId::operator=( const DetId& gen )
105 {
106    if (!gen.null() && ( gen.det()!=Ecal || gen.subdetId()!=EcalEndcap ))
107    {
108       throw cms::Exception("InvalidDetId");
109    }
110    id_ = gen.rawId();
111    return *this;
112 }
113
114 EEDetId
115 EEDetId::unhashIndex( int hi )
116 {
117    if( validHashIndex( hi ) )
118    {
119       const int iz ( hi<kEEhalf ? -1 : 1 ) ;
120       const uint32_t di ( hi%kEEhalf ) ;
121       const int ii ( ( std::upper_bound( kdi, kdi+(2*IY_MAX), di ) - kdi ) - 1 ) ;
122       const int iy ( 1 + ii/2 ) ;
123       const int ix ( kxf[ii] + di - kdi[ii] ) ;
124       return EEDetId( ix, iy, iz ) ;
125    }
126    else
127    {
128       return EEDetId() ;
129    }
130 }
131
```

```
132 int
133 EEDetId::ix( int iSC, int iCrys ) const
134 {
135   /*
136    *  ix() return individual crystal x-coordinate
137    *
138    *  Author    : B W Kennedy
139    *  Version   : 1.00
140    *  Created   : 21 December 2005
141    *  Last Mod  : 31 January 2006
142    *
143    *  Input     : iSC, iCrys - Supercrystal and crystal ids
144    */
145
146
147    int nSCinQuadrant = QuadColLimits[nCols];
148
149    if (iSC > 4*nSCinQuadrant || iSC < 1)
150    {
151       throw new std::exception();
152    }
153
154    //  Map SC number into (x>0,y>0) quadrant.
155    int iSCmap, iqx,iq;
156    if (iSC > 3*nSCinQuadrant)
157    {
158       iSCmap = iSC - 3*nSCinQuadrant;
159       iqx =  1;
160       iq=4;
161    }
162    else if (iSC > 2*nSCinQuadrant)
163    {
164       iSCmap = iSC - 2*nSCinQuadrant;
165       iqx = -1;
166       iq=3;
167    }
168    else if (iSC > nSCinQuadrant)
169    {
170       iSCmap = iSC - nSCinQuadrant;
171       iqx = -1;
172       iq=2;
173    }
174    else
175    {
176       iSCmap = iSC;
177       iqx = 1;
178       iq=1;
179    }
180
181    // Decide which column the SC is in
182    int iCol = 0 ;
183    while (iSCmap > QuadColLimits[iCol++]) ;
184    iCol-- ;
185
186    int ixCrys=-1;
187    if (iq == 1 || iq == 3)
188       ixCrys = iqx*(5*(iCol-1) + (int)(iCrys+4)/5);
189    else   if (iq == 2 || iq == 4)
190       ixCrys = iqx*(5*(iCol-1) + (iCrys-1)%5 + 1);
191
192    // returning a value from 1 to 100
193
194    return ixCrys;
195 }
196
197 int EEDetId::iy( int iSC, int iCrys ) const
198 {
```

```
199   /*
200    *  iy() return individual crystal y-coordinate
201    *
202    *  Author    : B W Kennedy
203    *  Version   : 1.00
204    *  Created   : 21 December 2005
205    *  Last Mod  : 31 January 2006
206    *
207    *  Input     : iSC, iCrys - Supercrystal and crystal ids
208    */
209
210    int nSCinQuadrant = QuadColLimits[nCols];
211    if (iSC > 4*nSCinQuadrant || iSC < 1)
212    {
213       throw new std::exception();
214    }
215
216    //  Map SC number into (x>0,y>0) quadrant
217    int iSCmap, iqy,iq;
218    if (iSC > 3*nSCinQuadrant)
219    {
220       iSCmap = iSC - 3*nSCinQuadrant;
221       iqy = -1;
222       iq=4;
223    }
224    else if (iSC > 2*nSCinQuadrant)
225    {
226       iSCmap = iSC - 2*nSCinQuadrant;
227       iqy = -1;
228       iq=3;
229    }
230    else if (iSC > nSCinQuadrant)
231    {
232       iSCmap = iSC - nSCinQuadrant;
233       iqy = 1;
234       iq=2;
235    } else
236    {
237       iSCmap = iSC;
238       iqy = 1;
239       iq=1;
240    }
241
242    // Decide which column the SC is in
243    int iCol = 0;
244    while (iSCmap > QuadColLimits[iCol++]) ;
245    iCol--;
246
247    int iSCy = iSCmap - QuadColLimits[iCol-1] + iYoffset[iCol];
248
249    int iyCrys=-1;
250    if (iq == 1 || iq == 3)
251       iyCrys = iqy*(5*(iSCy-1) + (iCrys-1)%5 + 1);
252    else if (iq == 2 || iq == 4)
253       iyCrys = iqy*(5*(iSCy-1) + (int)(iCrys+4)/5 );
254    return iyCrys;
255 }
256
257 int EEDetId::ixQuadrantOne() const
258 {
259    int iQuadrant = iquadrant();
260    if ( iQuadrant == 1 || iQuadrant == 4)
261       return (ix() - 50);
262    else if ( iQuadrant == 2 || iQuadrant == 3)
263       return (51 - ix());
264    //Should never be reached
265    return -1;
```

```
266 }
267
268 int EEDetId::iyQuadrantOne() const
269 {
270    int iQuadrant = iquadrant();
271    if ( iQuadrant == 1 || iQuadrant == 2)
272       return (iy() - 50);
273    else if ( iQuadrant == 3 || iQuadrant == 4)
274       return 51 - iy();
275    //Should never be reached
276    return -1;
277 }
278
279 int
280 EEDetId::iquadrant() const
281 {
282    if (ix()>50)
283    {
284       if(iy()>50)
285          return 1;
286       else
287          return 4;
288    }
289    else
290    {
291       if(iy()>50)
292          return 2;
293       else
294          return 3;
295    }
296    //Should never be reached
297    return -1;
298 }
299
300 int
301 EEDetId::isc() const
302 {
303    return isc( 1 + ( ix() - 1 )/nCrys,
304                1 + ( iy() - 1 )/nCrys ) ;
305 }
306
307 int
308 EEDetId::isc( int jx, int jy )
309 {
310    if( 0  < jx &&
311        21 > jx &&
312        0  < jy &&
313        21 > jy    )
314    {
315       const int iquad (  ( 10<jx && 10<jy ? 1 :
316                            ( 11>jx && 10<jy ? 2 :
317                              ( 11>jx && 11>jy ? 3 : 4 ) ) ) ) ;
318
319       const int iCol = ( 1 == iquad || 4 == iquad ? jx - 10 : 11 - jx ) ;
320       const int iRow = ( 1 == iquad || 2 == iquad ? jy - 10 : 11 - jy ) ;
321
322       static int nSCinQuadrant = ISC_MAX/4;
323
324       const int yOff ( iYoffset[iCol] ) ;
325
326       const int qOff ( nSCinQuadrant*( iquad - 1 ) ) ;
327
328       const int iscOne ( QuadColLimits[iCol-1] + iRow - yOff ) ;
329
330       return ( yOff                >= iRow   ? -1 :
331                ( QuadColLimits[iCol] <  iscOne ? -2 :
332                  iscOne + qOff ) ) ;
```

```
333    }
334    else
335    {
336       return -3 ; // bad inputs
337    }
338 }
339
340 int EEDetId::ic() const
341 {
342    /*
343     *  Return crystal number from (x,y) coordinates.
344     *
345     *  Author    : B W Kennedy
346     *  Version   : 1.00
347     *  Created   : 5 May 2006
348     *  Last Mod  :
349     *
350     *  Input     : ix, iy - (x,y) position of crystal
351     */
352
353    /*  Useful constants . */
354    int iQuadrant = iquadrant();
355    int icrCol=-1;
356    int icrRow=-1;
357
358    if (iQuadrant == 1 || iQuadrant == 3)
359    {
360       icrCol=(ixQuadrantOne()-1) % nCrys;
361       icrRow=(iyQuadrantOne()-1) % nCrys;
362    }
363
364    else if (iQuadrant == 2 || iQuadrant == 4)
365    {
366       icrRow=(ixQuadrantOne()-1) % nCrys;
367       icrCol=(iyQuadrantOne()-1) % nCrys;
368    }
369
370    int icrys = 5*icrCol + icrRow + 1;
371
372    return icrys;
373 }
374
375
376 bool
377 EEDetId::isNextToBoundary( EEDetId id )
378 {
379    return isNextToDBoundary( id ) || isNextToRingBoundary( id ) ;
380 }
381
382 bool
383 EEDetId::isNextToDBoundary( EEDetId id )
384 {
385    // hardcoded values for D boundary
386    return id.ix() == 50 || id.ix() == 51 ;
387 }
388
389
390 bool
391 EEDetId::isNextToRingBoundary(EEDetId id)
392 {
393    for (int i = -1; i <= 1; ++i) {
394       for (int j = -1; j <= 1; ++j) {
395          if ( ! validDetId( id.ix() + i, id.iy() + j, id.zside() ) ) {
396             return true;
397          }
398       }
399    }
```

```
400    return false;
401 }
402
403 int
404 EEDetId::iPhiOuterRing() const
405 {
406    int returnValue ( 0 ) ;
407    if( isOuterRing() )
408    {
409       const int ax ( abs( ix() - IX_MAX/2 ) ) ;
410       const int ay ( abs( iy() - IY_MAX/2 ) ) ;
411       returnValue = ax + 50 - ay ;
412       if( ay <= 47 ) --returnValue ;
413       if( ay <= 45 ) --returnValue ;
414       if( ay <= 42 ) --returnValue ;
415       if( ay <= 37 ) --returnValue ;
416       if( ay <= 35 ) --returnValue ;
417       if( ay <= 30 ) --returnValue ;
418       if( ay <= 25 ) --returnValue ;
419       if( ay <= 15 ) --returnValue ;
420       if( ay <= 10 ) --returnValue ;
421       const int iq ( iquadrant() ) ;
422       if( 1==iq )
423       {
424          returnValue = 91 - returnValue ;
425       }
426       else
427       {
428          if( 2==iq )
429          {
430             returnValue += 90 ;
431          }
432          else
433          {
434             if( 3==iq )
435             {
436                returnValue = 271 - returnValue ;
437             }
438             else
439             {
440                returnValue += 270 ;
441             }
442          }
443       }
444       returnValue = 1 + ( 360 + returnValue - 10 -1 )%360 ;
445    }
446 //   if( positiveZ() ) returnValue += 360 ;
447    return returnValue ;
448 }
449
450 EEDetId
451 EEDetId::idOuterRing( int iPhi , int zEnd )
452 {
453    iPhi -= 10 ; // phi=1 in barrel is at -10deg
454    while( iPhi <   1 ) iPhi+=360 ;
455    while( iPhi > 360 ) iPhi-=360 ;
456
457    const int index1 ( iPhi - 1 ) ;
458    const int quad   ( index1/90 ) ;
459    int       indexq (  index1 - quad*90 + 1 ) ;
460    if( 0==quad || 2==quad ) indexq = 91 - indexq ;
461    const int indexh ( indexq > 45 ? 91 - indexq : indexq ) ;
462    const int axh    ( indexh<=10 ? indexh :
463                       ( indexh<=12 ? 10 :
464                         ( indexh<=17 ? indexh - 2 :
465                           ( indexh<=18 ? 15 :
466                             ( indexh<=28 ? indexh - 3 :
```

```
467                                ( indexh<=30 ? 25 :
468                                  ( indexh<=35 ? indexh - 5 :
469                                    ( indexh<=39 ? 30 :
470                                      ( indexh<=44 ? indexh - 9 : 35 ))))))))) ;
471    const int ayh    ( indexh<=10 ? 50 :
472                       ( indexh<=12 ? 60 - indexh :
473                         ( indexh<=17 ? 47 :
474                           ( indexh<=18 ? 64 - indexh :
475                             ( indexh<=28 ? 45 :
476                               ( indexh<=30 ? 73 - indexh :
477                                 ( indexh<=35 ? 42 :
478                                   ( indexh<=39 ? 77 - indexh :
479                                     ( indexh<=44 ? 37 : 36 ))))))))) ;
480    const int bxh ( indexq>45 ? ayh : axh ) ;
481    const int byh ( indexq>45 ? axh : ayh ) ;
482    const int cx  ( ( quad==0 || quad==3 ? bxh : -bxh+1 ) + IX_MAX/2 ) ;
483    const int cy  ( ( quad==0 || quad==1 ? byh : -byh+1 ) + IY_MAX/2 ) ;
484
485    return EEDetId( cx, cy, ( zEnd > 0 ? 1 : -1 ) ) ;
486 }
487
488
489 EEDetId
490 EEDetId::offsetBy(int nrStepsX, int nrStepsY ) const
491 {
492         int newX = ix() + nrStepsX;
493         int newY = iy() + nrStepsY;
494
495         if( validDetId( newX, newY, zside() ) ) {
496                 return EEDetId( newX, newY, zside() );
497         } else {
498                 return EEDetId(0);
499         }
500 }
501
502 EEDetId
503 EEDetId::switchZSide() const
504 {
505         int newZSide = -1 * zside();
506         if( validDetId(ix(), iy(), newZSide ) ) {
507                 return EEDetId( ix(), iy(), newZSide );
508         } else {
509                 return EEDetId(0);
510         }
511 }
512
513 DetId
514 EEDetId::offsetBy( const DetId startId, int nrStepsX, int nrStepsY )
515 {
516         if( startId.det() == DetId::Ecal && startId.subdetId() == EcalEndcap ) {
517                 EEDetId eeStartId( startId );
518                 return eeStartId.offsetBy( nrStepsX, nrStepsY ).rawId();
519         } else {
520                 return DetId(0);
521         }
522 }
523
524 DetId
525 EEDetId::switchZSide( const DetId startId )
526 {
527         if( startId.det() == DetId::Ecal && startId.subdetId() == EcalEndcap ) {
528                 EEDetId eeStartId(startId);
529                 return eeStartId.switchZSide().rawId();
530         } else {
531                 return DetId(0);
532         }
533 }
```

```
534
535 bool
536 EEDetId::isOuterRing() const
537 {
538    const int kx ( ix() ) ;
539    const int ky ( iy() ) ;
540    const int ax ( kx>IX_MAX/2 ? kx-IX_MAX/2 : IX_MAX/2 + 1 - kx ) ;
541    const int ay ( ky>IY_MAX/2 ? ky-IY_MAX/2 : IY_MAX/2 + 1 - ky ) ;
542    return ( isOuterRingXY( ax, ay ) ||
543             isOuterRingXY( ay, ax )    ) ;
544 }
545
546 bool
547 EEDetId::isOuterRingXY( int ax, int ay )
548 {
549    return ( ( ax<=10 &&           ay==50 ) ||
550             ( ax==10 &&           ay>=48 ) ||
551             ( ax<=15 && ax>=11 && ay==47 ) ||
552             ( ax==15 &&           ay==46 ) ||
553             ( ax<=25 && ax>=16 && ay==45 ) ||
554             ( ax==25 &&           ay<=44 && ay>=43 ) ||
555             ( ax<=30 && ax>=26 && ay==42 ) ||
556             ( ax==30 &&           ay<=41 && ay>=38 ) ||
557             ( ax<=35 && ax>=31 && ay==37 ) ||
558             ( ax==35 &&           ay==36 )              ) ;
559 }
560
561 bool
562 EEDetId::validDetId(int crystal_ix, int crystal_iy, int iz)
563 {
564    bool valid = false;
565    if (crystal_ix < IX_MIN ||  crystal_ix > IX_MAX ||
566        crystal_iy < IY_MIN || crystal_iy > IY_MAX || abs(iz) != 1 )
567    {
568       return valid ;
569    }
570    if ( (crystal_ix >= 1 && crystal_ix <= 3 && (crystal_iy <= 40 || crystal_iy > 60) ) ||
571         (crystal_ix >= 4 && crystal_ix <= 5 && (crystal_iy <= 35 || crystal_iy > 65) ) ||
572         (crystal_ix >= 6 && crystal_ix <= 8 && (crystal_iy <= 25 || crystal_iy > 75) ) ||
573         (crystal_ix >= 9 && crystal_ix <= 13 && (crystal_iy <= 20 || crystal_iy > 80) ) ||
574         (crystal_ix >= 14 && crystal_ix <= 15 && (crystal_iy <= 15 || crystal_iy > 85) ) ||
575         (crystal_ix >= 16 && crystal_ix <= 20 && (crystal_iy <= 13 || crystal_iy > 87) ) ||
576         (crystal_ix >= 21 && crystal_ix <= 25 && (crystal_iy <= 8 || crystal_iy > 92) ) ||
577         (crystal_ix >= 26 && crystal_ix <= 35 && (crystal_iy <= 5 || crystal_iy > 95) ) ||
578         (crystal_ix >= 36 && crystal_ix <= 39 && (crystal_iy <= 3 || crystal_iy > 97) ) ||
579         (crystal_ix >= 98 && crystal_ix <= 100 && (crystal_iy <= 40 || crystal_iy > 60) ) ||
580         (crystal_ix >= 96 && crystal_ix <= 97 && (crystal_iy <= 35 || crystal_iy > 65) ) ||
581         (crystal_ix >= 93 && crystal_ix <= 95 && (crystal_iy <= 25 || crystal_iy > 75) ) ||
582         (crystal_ix >= 88 && crystal_ix <= 92 && (crystal_iy <= 20 || crystal_iy > 80) ) ||
583         (crystal_ix >= 86 && crystal_ix <= 87 && (crystal_iy <= 15 || crystal_iy > 85) ) ||
584         (crystal_ix >= 81 && crystal_ix <= 85 && (crystal_iy <= 13 || crystal_iy > 87) ) ||
585         (crystal_ix >= 76 && crystal_ix <= 80 && (crystal_iy <= 8 || crystal_iy > 92) ) ||
586         (crystal_ix >= 66 && crystal_ix <= 75 && (crystal_iy <= 5 || crystal_iy > 95) ) ||
587         (crystal_ix >= 62 && crystal_ix <= 65 && (crystal_iy <= 3 || crystal_iy > 97) ) ||
588         ( (crystal_ix == 40 || crystal_ix == 61) && ( (crystal_iy >= 46 && crystal_iy <= 55 )
589         ( (crystal_ix == 41 || crystal_ix == 60) && crystal_iy >= 44 && crystal_iy <= 57 ) ||
590         ( (crystal_ix == 42 || crystal_ix == 59) && crystal_iy >= 43 && crystal_iy <= 58 ) ||
591         ( (crystal_ix == 43 || crystal_ix == 58) && crystal_iy >= 42 && crystal_iy <= 59 ) ||
592         ( (crystal_ix == 44 || crystal_ix == 45 || crystal_ix == 57 || crystal_ix == 56) && c
593         ( crystal_ix >= 46 && crystal_ix <= 55 && crystal_iy >= 40 && crystal_iy <= 61 )
594       )
595    {
596       return valid;
597    }
598    valid = true;
599    return valid;
600 }
```

```
601
602 int EEDetId::distanceX(const EEDetId& a,const EEDetId& b)
603 {
604     return abs(a.ix()-b.ix());
605 }
606
607 int EEDetId::distanceY(const EEDetId& a,const EEDetId& b)
608 {
609   return abs(a.iy() - b.iy());
610 }
611
612 std::ostream& operator<<(std::ostream& s,const EEDetId& id)
613 {
614    return s << "(EE iz " << ((id.zside()>0)?("+ "):("- "))
615             << " ix " << id.ix() << " , iy " << id.iy() << ')';
616 }
```

-- DavidCockerill - 23-Aug-2010

This topic: Main > EEDetID
Topic revision: r1 - 2010-08-23 - DavidCockerill