```
lease check them!
3 Particle::TofCorrPID: offsetTof for single end: the input parameter are NOT correct! P
  lease check them!
9 Particle::TofCorrPID: offsetTof for single end: the input parameter are NOT correct! P
  lease check them!
0 Particle::TofCorrPID: offsetTof for single end: the input parameter are NOT correct! P
  lease check them!
1 Particle::TofCorrPID: offsetTof for single end: the input parameter are NOT correct! P
  lease check them!
2 Particle::TofCorrPID: offsetTof for single end: the input parameter are NOT correct! P
  lease check them!
3 Particle::TofCorrPID: offsetTof for single end: the input parameter are NOT correct! P
  lease check them!
4 Particle::TofCorrPID: offsetTof for single end: the input parameter are NOT correct! P
  lease check them!
5 Particle::TofCorrPID: offsetTof for single end: the input parameter are NOT correct! P
  lease check them!
6 Particle::TofCorrPID: offsetTof for single end: the input parameter are NOT correct! P
  lease check them!
7 Particle::TofCorrPID: offsetTof for single end: the input parameter are NOT correct! P
  lease check them!
8 Particle::TofCorrPID: offsetTof for single end: the input parameter are NOT correct! P
  lease check them!
9 Particle::TofCorrPID: offsetTof for single end: the input parameter are NOT correct! P
  lease check them!
0 Particle::TofCorrPID: offsetTof for single end: the input parameter are NOT correct! P
  lease check them!
1 Particle::TofCorrPID: offsetTof for single end: the input parameter are NOT correct! P
  lease check them!
2 Particle::TofCorrPID: offsetTof for single end: the input parameter are NOT correct! P
```

```
  double deltaT = -1000.0;
  if( ( ipmt>= 4 && barrel ) || ( ipmt!=7 && ipmt!=8 && !barrel ) || betaGamma<0.0 || a
bs(charge)!=1 || fabs(zrhit)>120.0 ) {
    cout << "Particle::TofCorrPID: offsetTof for single end: the input parameter are NO
T correct! Please check them!" << endl;
    return deltaT;
  }
```

```
    unsigned int ipmt = 0;
    if( readout ) {
      // barrel: 0:inner east, 1:inner west, 2:outer east, 3: outer west
      // endcap: 7:east endcap, 8:west endcap
      if( barrel ) { ipmt = ( ( st & 0xC0 ) >> 5 ) + ( ( ( st ^ 0x20 ) & 0x20 ) >> 5 )
 - 2; }
      else {
        if( !ismrpc ) {
          if( tofid[0]<=47 ) { ipmt = 7; }
          else { ipmt = 8; }
        }
        else {
          if( tofid[0]<=35 ) { ipmt = 7; }
          else { ipmt = 8; }
        }
      }
```

Only 703

- Bahbha 3.965 break
- Ditau 4410 break
- Data
  - 4600 MySQL
  - 4660 no successfully
  - 4840

- Bahbha 3.965 break : data copy
- Ditau 4410 break : n2gam
- Data
  - 4600 MySQL
  - 4660 no successfully : DTag
  - 4840

```
status = m_tuple1->addItem("nKs",                m_nKs,0,100);
status = m_tuple1->addIndexedItem("Mas_Ks",      m_nKs,m_Mas_Ks);
status = m_tuple1->addIndexedItem("Mom_Ks",      m_nKs,m_Mom_Ks);
status = m_tuple1->addItem("nPhi",               m_nPhi,0,100);
status = m_tuple1->addIndexedItem("Mas_Phi",     m_nPhi,m_Mas_Phi);
status = m_tuple1->addIndexedItem("Mom_Phi",     m_nPhi,m_Mom_Phi);
status = m_tuple1->addItem("n2gam",              m_n2gam,0,50000);
status = m_tuple1->addIndexedItem("Mas_2gam",    m_n2gam,m_Mas_2gam);
status = m_tuple1->addIndexedItem("Mom_2gam",    m_n2gam,m_Mom_2gam);
status = m_tuple1->addItem("nLambda0",           m_nLambda0,0,100);
status = m_tuple1->addIndexedItem("Mas_Lambda0",m_nLambda0,m_Mas_Lambda0);
status = m_tuple1->addIndexedItem("Mom_Lambda0",m_nLambda0,m_Mom_Lambda0);
```

```
// mass of 2 gammas: pi0 or eta
  double dltpi0=999.,mpi0=999.;
  m_n2gam = 0;
  if(nGam>=2&&nGam<40)
  {
    for(int i=0; i<nGam-1; i++)
    {
      for(int j=i+1; j<nGam; j++)
      {
        double masgg = (pGam[i]+pGam[j]).m();
        double momgg = (pGam[i]+pGam[j]).rho();
        m_Mas_2gam[m_n2gam] = masgg;
        m_Mom_2gam[m_n2gam] = momgg;
        m_n2gam++;
      }
    }
  }
  else
  {
    m_Mas_2gam[m_n2gam] = 0.0;
    m_Mom_2gam[m_n2gam] = 0.0;
  }
```