Argonne NATIONAL LABORATORY

... for a brighter future

U.S. Department of Energy

UChicago ▶ Argonne LLC

Office of Science U.S. DEPARTMENT OF ENERGY

A U.S. Department of Energy laboratory managed by UChicago Argonne, LLC

# ESD analysis of jets

*Belen Salvachua*

High Energy Physics Division
Argonne National Laboratory

**US ATLAS Analysis Jamboree: ANL**

**Tuesday 19th January 2010**

**http://indico.cern.ch/conferenceDisplay.py?confId=75247**

# Outline

- Why do you want to use ESD data for an analysis?
- In case you need to:  Where to start?
  - Useful web pages
  - Prepare your area
- Description of the code
- Where is the data?

# *Why do you want to use ESD data for analysis?*

do **NOT use** ESD data unless is totally needed!!!

**Reason NOT to use them**

AODs usually contain the information that you will need and they are easier to handle:

- Smaller in size

BIG ADVANTAGE!

- They usually require minor database access

**Reason to use them**

Sometimes you need information that is ONLY on ESD data

# *Where to start? Useful web pages*

- ATLAS computing twiki:
  https://twiki.cern.ch/twiki/bin/view/Atlas/AtlasComputing
- ATLAS Analysis Workbook twiki:
  https://twiki.cern.ch/twiki/bin/view/AtlasProtected/PhysicsAnalysisWorkBookRel15
- GoodRunList twiki:
  https://twiki.cern.ch/twiki//bin/viewauth/Atlas/GoodRunsLists
- AMI database:
  https://ami.in2p3.fr/AMI/servlet/net.hep.atlas.Database.Bookkeeping.AMI.Servlet.Command
- If working at Argonne ( ASC Workbook ):
  https://atlaswww.hep.anl.gov/twiki/bin/view/Workbook/WebHome

# ATLAS Computing Workbook

Analysis Workbook

## ATLAS Computing

Colour Key: *TWiki(unread)* *TWiki(read)* *Old php pages* *External Links*

**ATLAS Home**

**ATLAS TWiki**
Detectors
Trigger
Computing
Data Preparation
Physics
Help
Glossary

**ATLAS Computing**

**Problems**
Help
Running Jobs (Achelois)
Savannah
Using TWiki

**Communication**
HyperNews
Simba
Indico (Today)

**WorkBooks**
Computing
Physics Analysis
Software Development

**Tools**
AMI
Code Browsers
Tag Collector
AOD & ESD Contents

***Search***

### Getting Started

**Analysis WorkBook**
**Computing WorkBook**
For Newcomers
Help
Tutorials

### ATLAS e-mail

ATLAS e-mail management top page
ATLAS Hypernews List in e-group
Access to e-groups mail lists

### Other Communication

Collaborative Tools
Meetings (Today, All)
Savannah (about)

### Documentation

Computing TDR (pdf)
Documentation Management
Glossary

### Users and Developers

CernVM - Virtualization of ATLAS Software
AOD & ESD Contents
Athena
Core Software
Debugging Code
Event Data Model
Event Store
Info for Developers
Installing Software (advanced)
Release Recipes
**Software Development Workbook**
Trouble Running Jobs (Achelois)
Writing Code

### Activities

Combined Test Beam
FDR: Full Dress Rehearsal
Physics Validation
Software Validation
S/w Infrastructure Team (SIT)
Upgrade Simulation

### Tools

Doxygen (about)
Librarian Tools
Pacman (about)

### Code Management

AFS Directories
Code Distribution
Code Management
Nightly Builds (and ATN Testing)
Releases (and Project Builds)
Runtime Testing
Tag Collector (about)

### View Code

(help)
BNL Browser
Doxygen Classes (Search)
LXR
TagCollector
View SVN
ViewVC (Deprecated)

### Distributed Computing and Grid

AMI: Metadata Interface (about)
ADC: Distributed Computing
Databases
DDM: Distributed Data Management
Ganga
PanDA
AGIS
Pcache
Regional/Local Computing & Tier2
Tier0 Homepage
Tier1 Dataflow
Web Services

### Operations

CAF: CERN Analysis Facility
Computing Operations
Computing Operations
Database Deployment
Database Operations
Data Preparation
DDM Operations
Tier-0 Operations

# ATLAS Analysis Workbook

**Physics Analysis**
Combined Test Beam
Physics Validation
Physics Analysis Tools
Physics Analysis Workbook
Statistics Tools in ATLAS

**Physics Groups**
B Physics WG
Top WG
Standard Model WG
Higgs WG
SUSY WG
Exotics WG
Heavy Ions WG
Monte Carlo WG

**Combined Performance**
e/gamma
Flavor Tagging
Jet/EtMiss
Tau
Muon

**Other Groups**
Trigger Alg/Perf/Menu
InDet Tracking Perf

***Search***

## Preface Complete:

- Acknowledgements
- Using the Workbook
- Formatting Rules

## Introduction Complete:

- Getting Started
- Overview talks and articles on the ATLAS detector and Physics

## Data Formats Complete:

- Introduction
- ATLAS Event Data Model
- Contents of the RDO (Raw Data Object)
- Contents of the ESD (Event Summary Data) and AOD (Analysis Object Data)
- Contents of the Derived Physics Datasets (DPD)
- Contents of Root-tuples
- Contents of the TAGs

Examples

## Analysis Examples Complete:

- Introduction
- Important Information (on EDM changes in this release)
- Simple AOD analysis
- **NEW** AOD analysis with DataQuality flags, Luminosity and Trigger Information
- Accessing MC truth information
- Simple AOD Analysis - More exercises
- How to write an algorithm
- How to write an AlgTool
- Services in Athena
- Navigation (and BackNavigation)

# GoodRunList

Link to XML files

Athena packages and tags you need in your installed area

Tutorial

## Good Run Lists for Data

- Good run lists for data, see here.

## News

- 20091207: Update of tutorial will collisions grls.
- 20091126: Addition of section with good run lists for data.
- 20091127: Addition of mini tutorial for using good run lists with first data.

### 2009/12/07 - Recommended release and tags

The recommended release for the GoodRunsList package is `15.5.1` or `15.6.0` or greater. Please use the tags:

```
cmt co -r GoodRunsListsUser-00-00-08 DataQuality/GoodRunsListsUser
cmt co -r GoodRunsLists-00-00-67 DataQuality/GoodRunsLists
cmt co -r LumiBlockComps-00-00-78 LumiBlock/LumiBlockComps
cmt co -r CoolRunQuery-00-01-89 Database/CoolRunQuery
```

(This set of tags has been requested to go into release 15.6.0 and 16.0.0.)

The latest tag for the CoolRunQuery can be found at the runquery server, at the bottom center of the page.
In the tutorial we will work with the following tag:

```
CoolRunQuery-00-01-75
```

## Tutorial

The tutorial on the use of good run lists in physics analyses is split up in two parts.

1. The creation of and selection using good run lists is demonstrated here.
2. A follow-up tutorial demonstrates Luminosity Calculation, which uses good run lists as input.

For further good run list examples, one can always look in latest presentation(s) linked below

# GoodRunList XML files



## GoodRunListsForData

Not yet Certified as ATLAS Documentation

- Recorded luminosity blocks with collisions
- Beams Lists

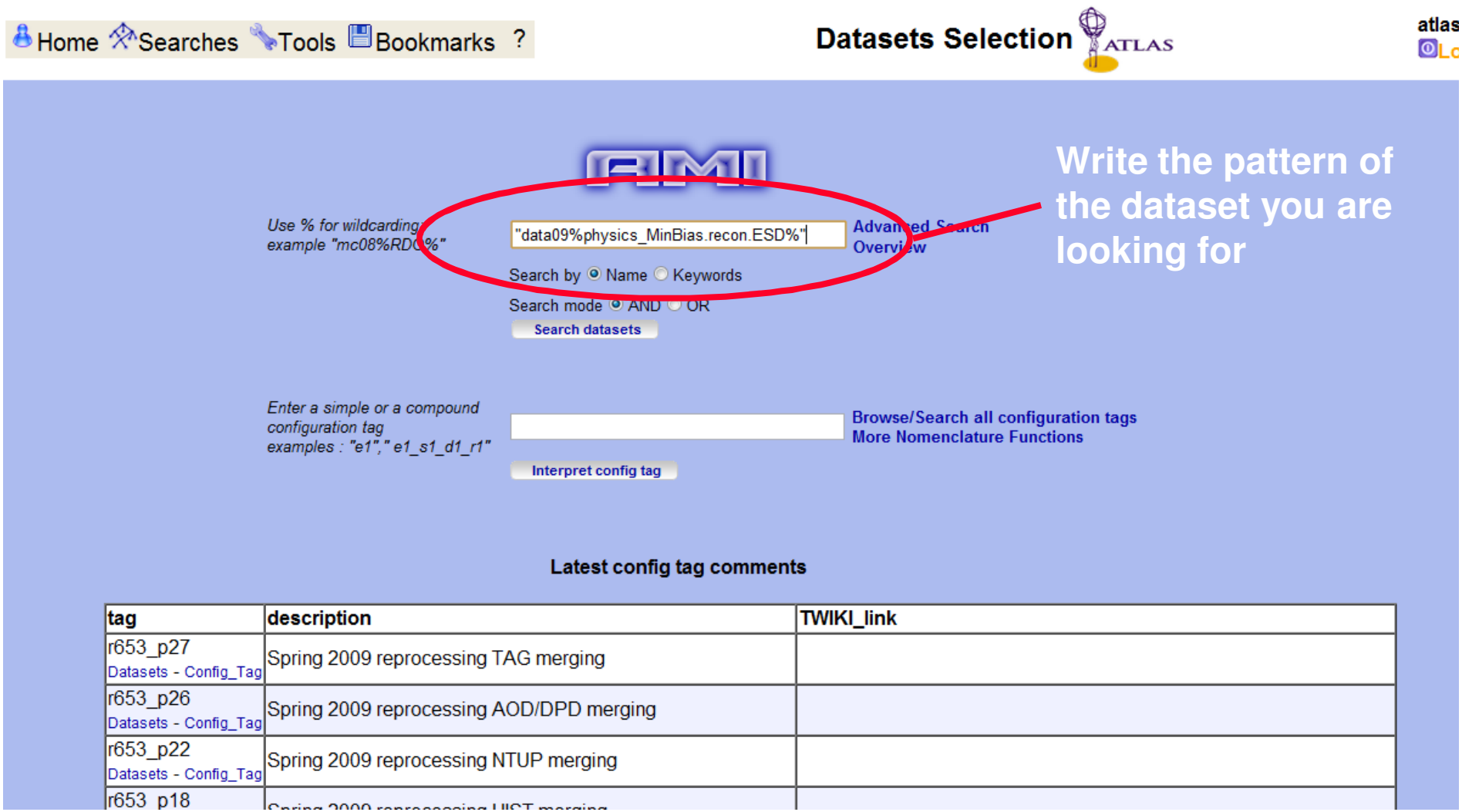
This page gives an overview of the available good run lists for data.
The good-run list portal, with documentation and an extended tutorial how to use good-run lists in physics analysis, is found in GoodRunsLists.
For those with little time: a mini tutorial can be found in GoodRunsListsTutorial.

## Recorded luminosity blocks with collisions

Information has (sofar) been taken from InterestingRuns2009.
Only luminosity blocks **that contain collisions** are selected in the good run lists.
An empty field means undefined, as in: the logical OR of all possible options.

| Run no. | Beam condictions | good run list | Special remarks | Beams | Solenoid | PIX/SCT |
|---|---|---|---|---|---|---|
| all runs untill 142383 | 900 GeV collisions | collisions_stablebeams_minbias_900GeV.xml | see here for details | Stable | | |
| all runs untill 142383 | 900 GeV collisions | collisions_stablebeams_magsol_minbias_900GeV.xml | see here for details | Stable | On | |
| all runs untill 142383 | 900 GeV collisions | collisions_stablebeams_magsol_pixsctgreen_minbias_900GeV.xml | see here for details | Stable | On | Green |
| all runs untill 142383 | 900 GeV collisions | collisions_stablebeams_pixsctgreen_minbias_900GeV.xml | see here for details | Stable | | Green |
| run 142308, 142402 | 2 TeV collisions | collisions_minbias_2TeV.xml | see here for details | Unstable | | |
| run 142308, 142402 | 2 TeV collisions | collisions_magsol_minbias_2TeV.xml | see here for details | Unstable | On | |

# AMI database




Write the pattern of the dataset you are looking for

# ANL ASC Workbook

Argonne NATIONAL LABORATORY

Jump

**Workbook**

<< Back to ANL ASC

Setting Up Account
Running Athena
EventDisplay
Using Pathena
Grid data access
CVS/SVN access
Using PC farm (PCF)
PCF local data
Printing at ASC

TWiki > Workbook Web > WebHome (25 Nov 2009, SergeiChekanov)

## Welcome to ANL ASC Computer Workbook

<< Back to ANL ASC

- Setting Up Account
- Running Athena
- EventDisplay
- Using Pathena
- Grid data access
- CVS and SVN access
- Using PC farm (PCF)
- PCF local data
- Printing at ASC

Edit | Attach | Print version | History: r10 < r9 < r8 < r7 < r6 | Backlinks | Raw View | Raw edit | More topic actions

Topic revision: r10 - 25 Nov 2009 - 20:08:30 - SergeiChekanov

# Let's start working…

- Log on into atlas1.hep.anl.gov (for example)
- Setup Athena release 15.6.1
  - cd /users/belen/workarea/Data09_900GeV/15.6.1 (use your testarea instead)
  - source s15.6.1.sh

```
source /share/grid/app/asc_app/asc_rel/1.0/setup-script/set_asc.sh #set condor
export AVERS=15.6.1                    # set atlas release here
export TEST_AREA=$HOME/workarea/Data09_900GeV    # your work area
source /share/grid/app/asc_app/asc_rel/1.0/setup-script/set_atlas.sh #set athena
```

- Download all needed packages and compile them:

```
kinit belen@CERN.CH
### AnalysisSkeleton: Template used in our analysis
cmt co PhysicsAnalysis/AnalysisCommon/UserAnalysis
### Some additional examples (optional)
#cmt co PhysicsAnalysis/AnalysisCommon/AnalysisExamples/
### GoodRunList (Notice different tags than recommended on thetwiki),
cmt co -r GoodRunsLists-00-00-67 DataQuality/GoodRunsLists
cmt co -r GoodRunsListsUser-00-00-08 DataQuality/GoodRunsListsUser
cmt co -r LumiBlockComps-00-00-78 LumiBlock/LumiBlockComps
cmt co -r CoolRunQuery-00-01-89 Database/CoolRunQuery
### Compile each package by doing:
cd $TEST_AREA/$AVERS/PhysicsAnalysis/AnalysisCommon/UserAnalysis/cmt
cmt config && gmake
```

# Prepare your Analysis Code I: Using AnalysisSkeleton

- Similar to what was explained for AODs, only difference is that we'll access cells inside the jets You can add this to the DragonFly class ( explained by Esteban)
- Otherwise you can start from scratch: Make a copy of
  `PhysicsAnalysis/AnalysisCommon/UserAnalysis/src/AnalysisSkeleton.cxx`
  `PhysicsAnalysis/AnalysisCommon/UserAnalysis/UserAnalysis/AnalysisSkeleton.h`
- Create a new class similar to AnalysisSkeleton ( I called it MyJetAnalysis )
- Add it to:
  `PhysicsAnalysis/AnalysisCommon/UserAnalysis/src/components/UserAnalysis_entries.cxx`

```
#include "UserAnalysis/AnalysisSkeleton.h"
#include "UserAnalysis/MyJetAnalysis.h"

#include "GaudiKernel/DeclareFactoryEntries.h"
DECLARE_ALGORITHM_FACTORY( AnalysisSkeleton )
DECLARE_ALGORITHM_FACTORY( MyJetAnalysis )

DECLARE_FACTORY_ENTRIES( UserAnalysis ) {
  DECLARE_ALGORITHM( AnalysisSkeleton )
  DECLARE_ALGORITHM( MyJetAnalysis )
}
```

# Prepare your Analysis Code II

- Modify your requirements file if you add dependencies to other libraries

```
use JetUtils                    JetUtils-*                    Reconstruction/Jet
```

- We are going to create some histograms with:

  - Number of jets
  - Jet pT
  - Jet eta
  - Jet phi

Only accessible through ESDs

- Number of cells inside a jet
- Jet energy per layer (EM and HAD scale)
- Jet cell energy at EM scale
- Distance between cell and jet axis
- Jet cell energy density

# Prepare your Analysis Code III

```
StatusCode MyJetAnalysis::CBNT_initialize() {
                    . . . . .
  //////////////////////
  m_h_jet_njet = new TH1F("jet_njet","jet_njet",20,0,10);
  sc = m_thistSvc->regHist("/AANT/Jet/jet_njet",m_h_jet_njet);

  m_h_jet_pt = new TH1F("jet_pt","jet_pt",100,-10,200);
  sc = m_thistSvc->regHist("/AANT/Jet/jet_pt",m_h_jet_pt);
                    . . . . .
```

Histograms for general properties of the jets

Histograms of jet cell energy for each calorimeter layer

```
for (int ilay=0; ilay<24; ilay++){
    // Cell energy at EM scale
    sprintf(name,"jet_cell_eem[%d]",ilay);
    m_h_jet_cell_eem[ilay] = new TH1F(name,"Layer: "+(*m_LayerName[ilay]),200,-10,100);
    sprintf(name,"/AANT/Jet/jet_cell_eem[%d]",ilay);
    sc = m_thistSvc->regHist(name,m_h_jet_cell_eem[ilay]);

    // Cell deltaR( jet- cell)
    sprintf(name,"jet_cell_deltaR[%d]",ilay);
    m_h_jet_cell_deltaR[ilay] = new TH1F(name,"Layer: "+(*m_LayerName[ilay]),200,0, 2);
    sprintf(name,"/AANT/Jet/jet_cell_deltaR[%d]",ilay);
    sc = m_thistSvc->regHist(name,m_h_jet_cell_deltaR[ilay]);
                    . . . . .
}
```

# Prepare your Analysis Code IV

```
/// Method to fill Jet histograms
StatusCode MyJetAnalysis::fillJetHistograms() {
  StatusCode sc;
  MsgStream mLog( messageService(), name() );

  // ----  retrieve jets ---------------
  const JetCollection * jetTES = 0;
  sc = m_storeGate->retrieve( jetTES, m_jetContainerName);
  if( sc.isFailure()  ||  !jetTES ) {
    mLog << MSG::FATAL << "No Jet container found in TDS" << endreq;
    return StatusCode::FAILURE;}
   if (jetTES->size() < 1) return StatusCode::SUCCESS;

 // ---- fill histograms ---------------
  m_h_jet_njet->Fill(jetTES->size());

 // ---------- LOOP over JETS -------------------------
  JetCollection::const_iterator jetItr  = jetTES->begin();
  JetCollection::const_iterator jetItrE = jetTES->end();
  for (; jetItr != jetItrE; ++jetItr) {
    HepLorentzVector p4((*jetItr)->px(),(*jetItr)->py(),(*jetItr)->pz(),(*jetItr)->e());
    HepLorentzVector p4_em ( (*jetItr)->constituent_sum4Mom() );

    //  Fill histograms: Jet pT, eta and phi
    m_h_jet_pt->Fill(p4.perp()/GeV);
    m_h_jet_eta->Fill(p4.eta());
    m_h_jet_phi->Fill(p4.phi());

      . . . . . Continue in next page . . . . .
```

Athena Error Reporting
https://twiki.cern.ch/twiki/bin/view/Atlas/ReportingErrors

# Prepare your Analysis Code V

```
// ------- LOOP over cells -----------------
 NavigationToken<CaloCell,double> cellToken;
 (*jetItr)->fillToken(cellToken,double(1.));
 NavigationToken<CaloCell,double>::const_iterator  cbeg = cellToken.begin();
 NavigationToken<CaloCell,double>::const_iterator  cend = cellToken.end();
 mLog << MSG::DEBUG << "Get cell token" <<endreq;
 const double mm3 = millimeter*millimeter*millimeter;
 int ncells = 0;
 double jet_energy_from_cells_emscale = 0.;
 for (; cbeg != cend; ++cbeg)
   {
     ncells++;
     const CaloCell* thisCell = *cbeg;
     const CaloSampling::CaloSample s = CaloSampling::getSampling( *thisCell );
     double cell_weight = cellToken.getParameter(cbeg);
     double cell_eta = thisCell->eta();
     double cell_phi = thisCell->phi();
     double cell_energy  = (thisCell->e())/GeV;
     double cell_wenergy = cell_energy*cell_weight;
     double cell_volume  = thisCell->caloDDE()->volume()/mm3;
     // Fill histograms: Cell energy, DeltaR(cell-jet), Cell energy density
     m_h_jet_cell_eem[s]->Fill( cell_wenergy );
     double deta = p4.eta() - cell_eta;
     double dphi = fabs(JetDistances::deltaPhi(p4.phi(),cell_phi));
     double deltaR = sqrt(std::pow( deta , 2 ) + std::pow( dphi , 2 ));
     m_h_jet_cell_deltaR[s]->Fill( deltaR );
       . . . . . .
}
```

# Prepare your job options

- I use a top options that includes RecExCommon:

```
EvtMax=-1 ## number of event to process
## include your algorithm job options here ##########################
UserAlgs=[ "GoodRunsListsUser_oneSelection_Belen.py",
           "MyJetAnalysis_jobOptions.py"
           ]
##### SELECT INPUT DATA ##############################################
from glob import glob
INPUTFILES = glob("/data1/chakanau/data/data09_900GeV.*/ESD*pool.root*")
 . . . And some lines more . . .

## Read settings for performance DPD set ESD to true
readRDO = False
readESD = True
readAOD = False
 . . . And some more . . .

## main jobOption - must always be included
include ("RecExCommon/RecExCommon_topOptions.py")
```

# Where is the data at ANL?

- Runs inside good run list: https://twiki.cern.ch/twiki/bin/view/AtlasProtected/GoodRunListsForData

  *collisions_stablebeams_minbias_900GeV.xml*

- Are in: atlas1@/data1/chakanau/data/
- Also in PC farm: /data2/data09_900GeV/MinBias.merge/ESD/r988

```
data09_900GeV.00141748.physics_MinBias.recon.ESD.r988_tid101462_00
data09_900GeV.00141749.physics_MinBias.recon.ESD.r988_tid101466_00
data09_900GeV.00141811.physics_MinBias.recon.ESD.r988_tid101484_00
data09_900GeV.00141994.physics_MinBias.recon.ESD.r988_tid101490_00
data09_900GeV.00141999.physics_MinBias.recon.ESD.r988_tid101502_00
data09_900GeV.00142065.physics_MinBias.recon.ESD.r988_tid101519_00
data09_900GeV.00142149.physics_MinBias.recon.ESD.r988_tid101528_00
data09_900GeV.00142154.physics_MinBias.recon.ESD.r988_tid101539_00
data09_900GeV.00142165.physics_MinBias.recon.ESD.r988_tid101568_00
data09_900GeV.00142166.physics_MinBias.recon.ESD.r988_tid101577_00
data09_900GeV.00142171.physics_MinBias.recon.ESD.r988_tid101580_00
data09_900GeV.00142174.physics_MinBias.recon.ESD.r988_tid101587_00
data09_900GeV.00142189.physics_MinBias.recon.ESD.r988_tid105233_00
data09_900GeV.00142190.physics_MinBias.recon.ESD.r988_tid101603_00
data09_900GeV.00142191.physics_MinBias.recon.ESD.r988_tid101608_00
data09_900GeV.00142193.physics_MinBias.recon.ESD.r988_tid101619_00
data09_900GeV.00142195.physics_MinBias.recon.ESD.r988_tid105211_00
data09_900GeV.00142383.physics_MinBias.recon.ESD.r988_tid101673_00
```

# *Time to try it . . .*

- Some ESD files are also available from atlas17 and altas16:
  - /data/nfs/common/data900/ESD