

Usage of DQ flags in Physics Analysis

Max Baak, CERN
(on behalf of Data Quality group,
with thanks to LumiCalc group)



LBL ATLAS Analysis Jamboree
3 November '09

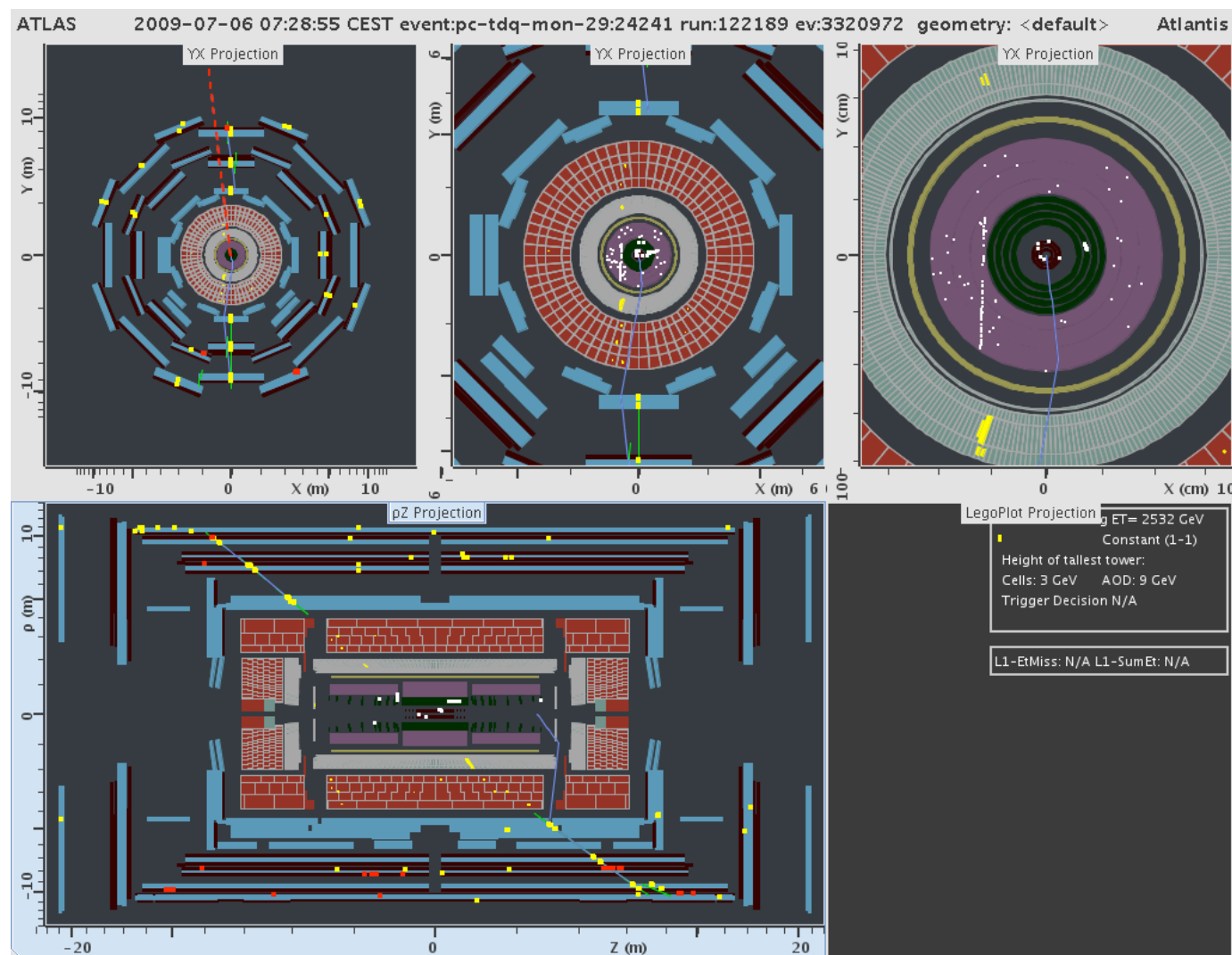


Outline

- Introduction to DQ Status Flags
- Storage and Access of DQ flags
 - Atlas Run Query tool
 - GoodRunsLumiBlock lists
- Overview of Good-Run-List tutorial
Use of GoodRuns-LumiBlock lists in your analysis
- Prospects/Conclusions

Data Quality Assurance

- Cosmic ray muon in MS and ID ...



- Q: Good track?
- Q: Good LB?
- Q: Good Run?

- A: Data Quality Assurance

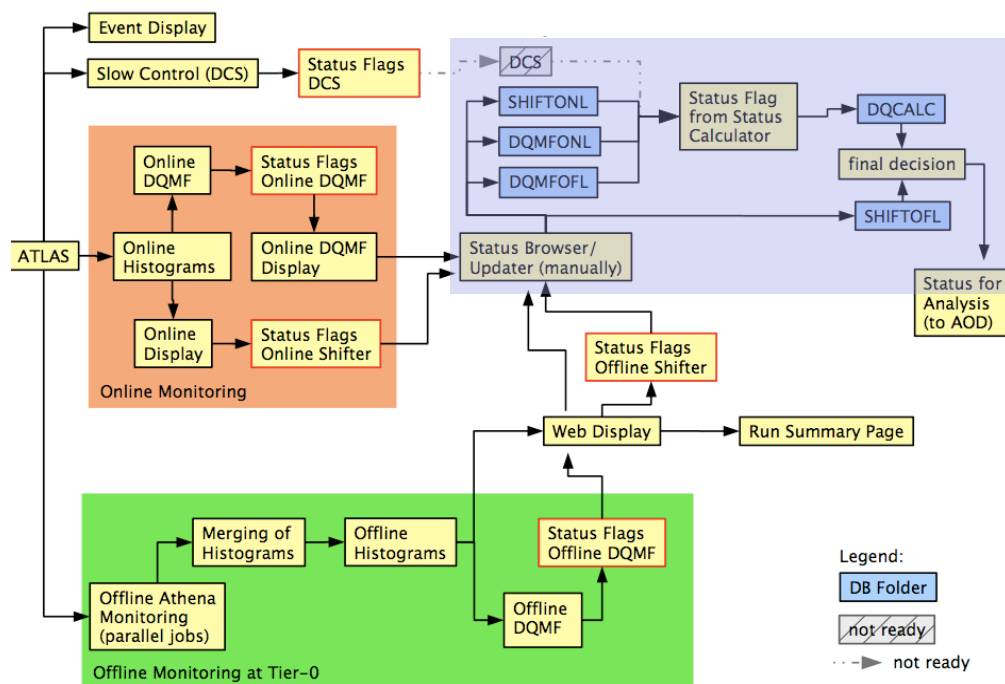
ATLAS Data Quality Monitoring

DQ group coordinated by Mario Martinez-Perez and Michael Hauschild

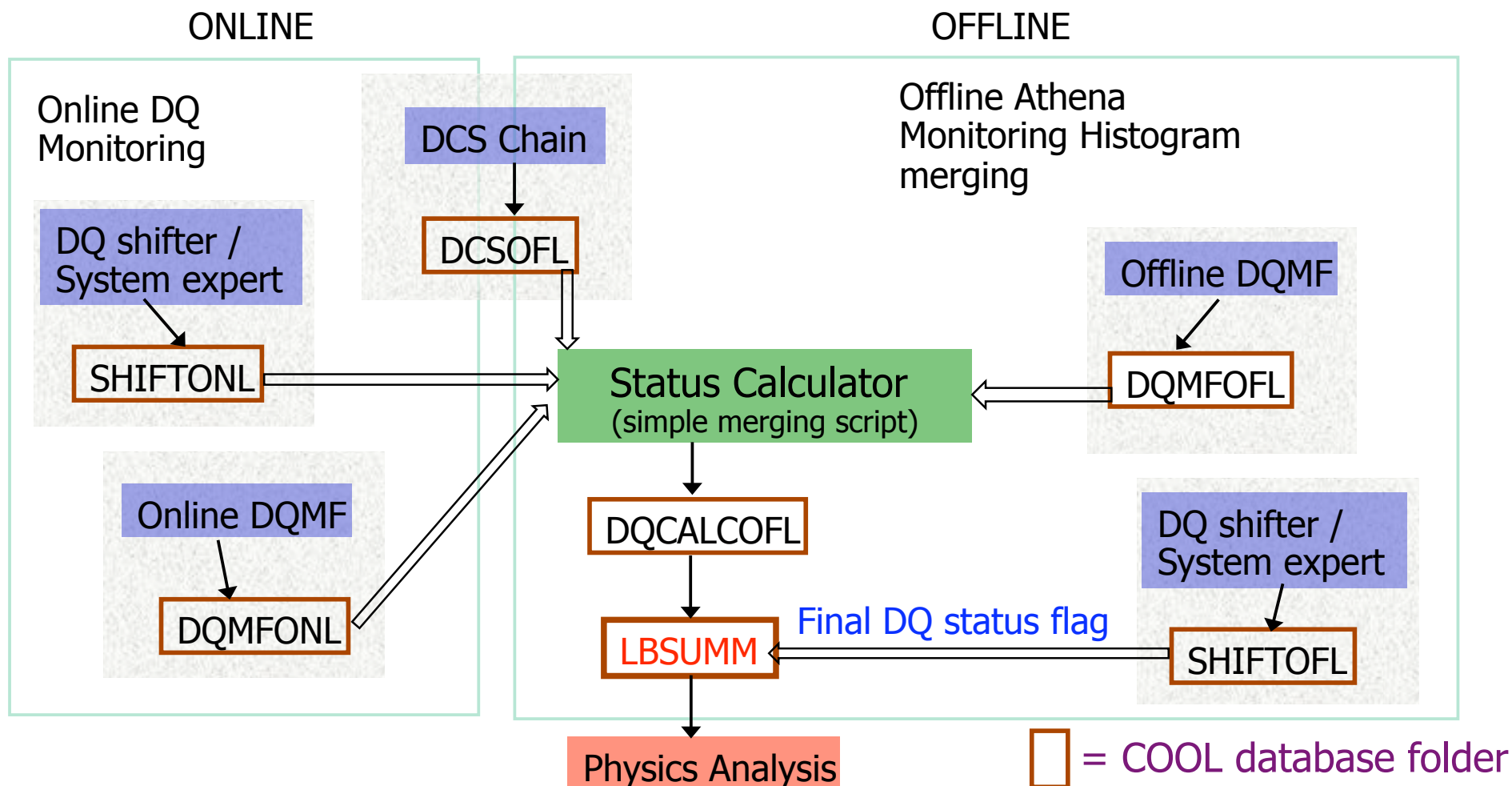
- Responsible for on- and offline DQ monitoring and assessment.
- Weekly DQ meeting, daily for data taking
- Representatives from all detector systems and CP groups
- Documentation:
 - Main TWiki: <https://twiki.cern.ch/twiki/bin/view/Atlas/DataQuality>
 - All DQ links: <http://atlasdqm.web.cern.ch/atlasdqm/>
 - Hypernews: <https://groups.cern.ch/group/hn-atlas-DataQuality/default.aspx>

Extensive monitoring available, including automated checks

- Many ingredients to entire DQ chain
 - Online, offline, merging and storages
- Online monitoring (also remote)
 - Real-time online monitoring
 - Athena-based monitoring (incl. event displays) using full-reconstruction
 - Slow control: Detector Control System
- Offline monitoring
 - Athena-based monitoring of all streams at Tier-0/1 reconstruction, frequent updates.



Data Quality Assessment



The monitoring information is concentrated into “DQ status flags”

- Assigned per lumi-block (~1min), not trigger stream aware
- Simple detector-geometrical separation
- Several folders for each DQ flag in conditions database (COOL)

Data Quality Status Flags

- DQ flag: traffic light → reflects overall DQ status of subdetector



- DQ status flags
 - Detector subsystem (assigned to barrel & end-caps, and some by task)
 - Trigger (per L1 system and trigger slice)
 - Combined performance groups (3 per group)
 - No DQ flags for physics groups/analyses yet (see later in talk ...)
- Additional status flags (incl. quantitative info) to indicate on- and offline event losses
 - L1 deadtime
 - Failed events in DAQ/HLT → debug stream
 - Event losses in Tier-0/1 (re-)reconstruction (failed files and/or event losses)
- Documentation (TWiki R. Hawkings):
 - <https://twiki.cern.ch/twiki/bin/view/Atlas/CoolDetStatus>
 - [DetectorDescription/DetDescrCond/DetectorStatus/doc/mainpage.h](https://twiki.cern.ch/twiki/bin/view/Atlas/DetectorDescription/DetDescrCond/DetectorStatus/doc/mainpage.h)

Data Quality flags in physics analysis

- Q: Why DQ flags?
- A: Mechanism to ensure that results of DQ and combined performance groups can be applied consistently to physics analyses!
- Q: What DQ flags are out there?
- A: Quite a few! See next slides ;-)

Current DQ flags in COOL (1)

101 PIXB	Pixel barrel
102 PIX0	Pixel B-layer
104 PIXEA	Pixel endcap A
105 PIXEC	Pixel endcap C
111 SCTB	SCT barrel
114 SCTEA	SCT endcap A
115 SCTEC	SCT endcap C
121 TRTB	TRT barrel
124 TRTEA	TRT endcap A
125 TRTEC	TRT endcap C
130 IDGL	ID global
140 IDAL	ID alignment
150 IDBS	Beamspot
160 IDPF	ID performance
170 IDBCM	ID Beam conditions monitor

202 EMBA	LAR EM barrel +z
203 EMBC	LAR EM barrel -z
204 EMECA	LAR EM endcap A
205 EMECC	LAR EM endcap C
214 HECA	LAR HEC endcap A
215 HECC	LAR HEC endcap C
224 FCALA	LAR FCAL endcap A
225 FCALC	LAR FCAL endcap C
230 TIGB	Tile global
232 TILBA	Tile barrel +z
233 TILBC	Tile barrel -z
234 TIEBA	Tile extended barrel A
235 TIEBC	Tile extended barrel C
244 MBTSA	MBTS trigger scintillator side A
245 MBTSC	MBTS trigger scintillator side C
251 CALB	Calorimeter global barrel
254 CALEA	Calorimeter global endcap A
255 CALEC	Calorimeter global endcap C

- Main detector systems – (mostly) primary flags

Current DQ flags in COOL (2)

302 MDTBA	MDT barrel +z
303 MDTBC	MDT barrel -z
304 MDTEA	MDT endcap A
305 MDTEC	MDT endcap C

312 RPCBA	RPC barrel A
313 RPCBC	RPC barrel C

324 TGCEA	TGC endcap A
325 TGCEC	TGC endcap C

334 CSCEA	CSC endcap A
335 CSCEC	CSC endcap C

350 LCD	LUCID global
353 LCDA	LUCID side A
354 LCDC	LUCID side C
360 ALFA	ALFA luminosity monitor
370 ZDC	ZDC luminosity monitor

- Main detectors – mostly primary DQ flags

401 L1CAL	L1 calorimeter trigger
402 L1MUB	L1 muon trigger barrel
403 L1MUE	L1 muon trigger endcaps
404 L1CTP	L1 central trigger

411 TRCAL	Calorimeter trigger
-----------	---------------------

421 TRBJT	trigger b-jet slice
422 TRBPH	trigger B-physics slice
423 TRCOS	trigger cosmics slice
424 TRELE	trigger electron slice
425 TRGAM	trigger gamma slice
426 TRJET	trigger jet slice
427 TRMET	trigger muon slice
428 TRMBI	trigger min bias slice
429 TRMUO	trigger muon slice
430 TRTAU	trigger tau slice
431 TRIDT	trigger ID tracking

450 LUMI	Luminosity OK (e.g. no missing files from SFO)
----------	---

460 RUNCLT	Run-control automatic flag
461 RCOPS	Run-control operator flag

- Trigger – primary and secondary DQ flags.

Current DQ flags in COOL (3)

501 EIDB electron ID barrel
504 EIDEA electron/photon ID endcap A
505 EIDAC electron/photon ID endcap C

511 MIDB muon ID barrel
514 MIDEA muon ID endcap A
515 MIDEA muon ID endcap C

521 JETB jet barrel
524 JETEA jet endcap A
525 JETEC jet endcap C

530 MET missing ET

541 BTGB b-tagging barrel
544 BTGEA b-tagging endcap A
545 BTGEC b-tagging endcap C

551 TAUB tau barrel
554 TAUEA tau endcap A
555 TAUEC tau endcap C

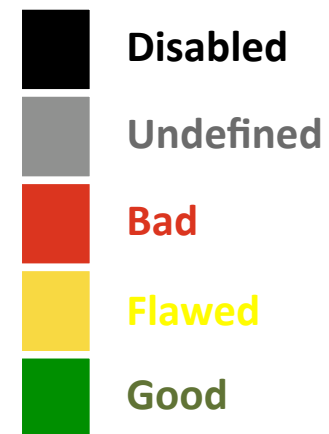
- CP groups – secondary DQ flags.

- In terms of DQ, sub-detector granularity determines definition of good physics objects
 - Eg. Only central or forward good electrons
- Definition of secondary DQ status flags in process of being finalized by CP groups (offline)
 - Perhaps also online by Global Monitoring
- **Open to feedback!**

Data Quality flag values

- DQ traffic light system explained:

- **Black (-1):** disabled Subdetector is disabled
- **Grey (0):** undefined Very short runs, or problems DQ monitoring. Decided upon later.
- **Red: (1)** bad Not good for physics
- **Yellow (2):** flawed Use with caution. Decided upon later.
- **Green (3):** good Good for physics use.

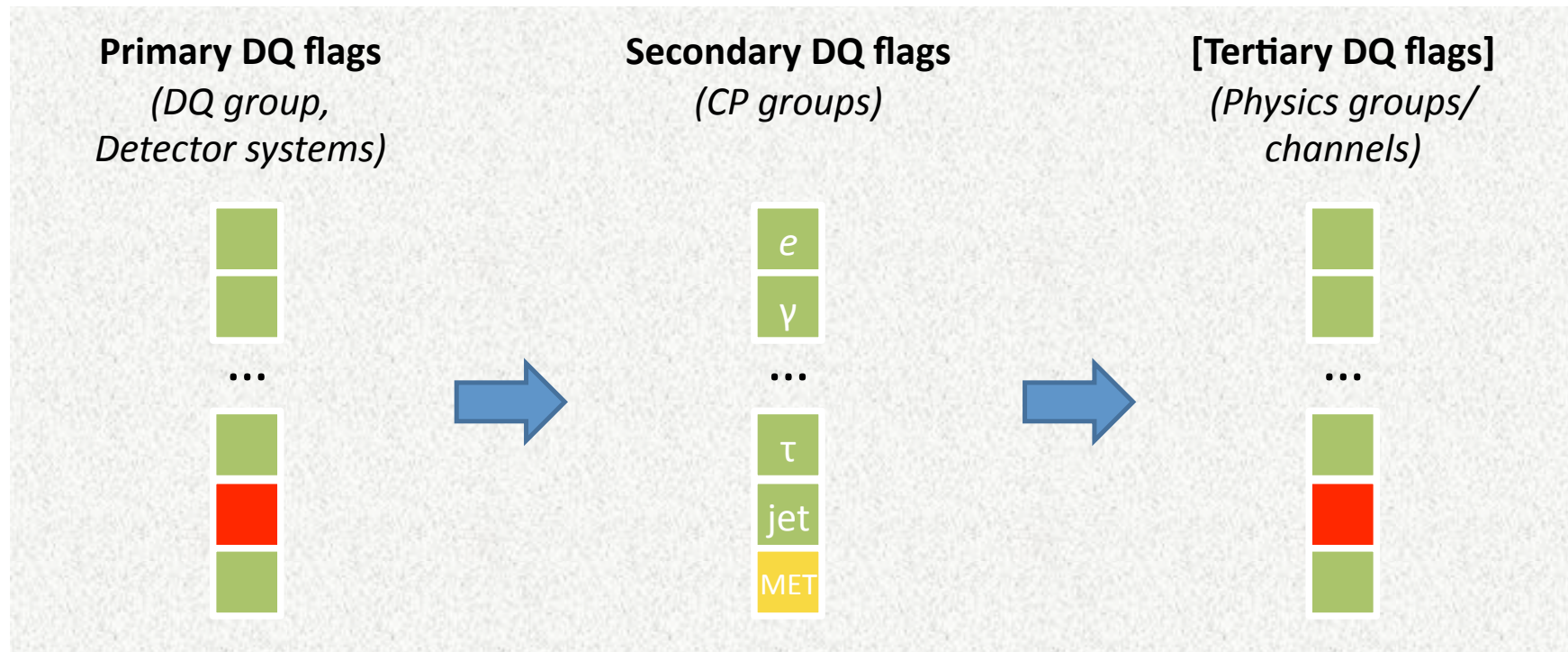


- Order: black < grey < red < yellow < green

- May also set "dead fraction" and "thrust" (floats) for each sub-detector

DQ flags hierarchy

- DQ flags hierarchy consists of three levels:



- Yellow and grey flags indicate that work (eg. calibration) is still ongoing. **Eventually no yellow or grey flags.**
- Yellow flags allowed. Usage depends on physics analysis.
- In process of being finalized.
- **No yellow flags for physics analysis!**
- (Tertiary flags not yet defined.)

Good-Runs-LumiBlocks lists

- Responsibility of performance groups to determine combination of primary DQ flags that define good physics object.
 - Stored as **secondary DQ flags** in COOL
 - Eg. Electron, photon, jet, MET, etc.
- Based on final states (combo of good physics objects), physics groups determine query of DQ flags employed in each physics analysis.
 - Decision may be stored as **tertiary DQ flags** in COOL per physics channel.

➔ This combination of physics DQ flags, together with other criteria (eg. trigger selection, run-range), applied to list of valid runs, determines Good-Runs/Luminosity-Blocks list.

- Physics channel dependent lists.

Storage of Data Quality flags

- The primary storage medium is conditions DB (COOL)
 - Once systems and groups have finalized DQ assessment → tag and lock DB
- During first-pass reconstruction (at Tier0), available DQ flags are copied from COOL into meta-data store.
 - Available in (d)ESD, (d)AOD, DnPD, TAG files
 - W/o DQ assessment from express stream: only online DQ info available
 - W/ DQA from express stream: also offline DQ flags
- After first-pass reconstruction, DQ flags are refined and updated in COOL
 - Values in derived files may be out-of-date after DQ update. (except Tag DB)
 - While DQ flags should be stable in time, this cannot be guaranteed
 - **Do not use DQ information in derived files for physics! (only for debugging)**
 - [Updates only reflected in metadata of (d)ESD/(d)AOD after reprocessing, O(months) later]

Data Quality flags in physics analysis

- Why DQ flags? A: Mechanism to ensure that results of DQ and combined performance groups can be applied consistently to physics analyses!
- Filtering on Data Quality flags done either in user analysis, or ntuples.
 - Ie. No Filtering applied at the making of (d)ESD or (d)AOD.

Approaches to use Data Quality information in physics analyses:

1. Access tertiary DQ flags via Athena job directly from COOL
 - This requires either direct Oracle access.
 - On the grid this would mean access to the closest Tier-1 Oracle server.
 - Not recommended until COOL caching works perfectly. ('Squid/Frontier')
 - Use extracted SQLite DB files instead

2. Use the Tag DB as primary event selection medium

Not discussed
in this tutorial

3. Use of dedicated Good Run-Lumi Block lists (pre-job query step)
 - (Only one sofar) perfectly integrated with Luminosity calculation.

Many tools to browse DQ information

- Collection of tools available to set/query/merge DQ flags in COOL
 - https://twiki.cern.ch/twiki/bin/view/Atlas/CoolDetStatus#Standalone_tools
 - detStatus_[query, merge, set, upload] stand-alone python scripts
- DQ status browser
 - <http://atlasdqm.web.cern.ch/atlasdqm/DQBrowser/DBQuery.html>
 - Browse and set DQ flags
- RunSummary Tool – “Shaun Roe’s run-summary page”
 - <http://atlas-service-db-runlist.web.cern.ch/atlas-service-db-runlist/query.html>
- dumpFileMetaData.py – quick and simple
- Access in Athena: DetectorStatus Package (python/C++)
 - http://atlas-computing.web.cern.ch/atlas-computing/links/nightlyDevDirectory/AtlasOffline/latest_doxygen/InstallArea/doc/DetectorStatus/html/index.html
- Atlas Run Query (AtIRunQuery) – user-friendly
 - <http://atlas-runquery.cern.ch/>
- GoodRunsList tools (using AtIRunQuery & DetStatusSvc)
 - See following slides ...

- ATLAS Run Query – Example: Cosmics run search with DQ flags

Search Result

Selection rule: find run 90270-90350 and events 100k+ / show run and ev and dq sct,rt,lar

Query command: AtIRunQuery.py --run "90270-90350" --events "100000+" --show run --show events --show "dq SCTB SHIFTOFL" --show "dq SCTEA SHIFTOFL" --show "dq SCTEC SHIFTOFL" --show "dq TRTB SHIFTOFL" --show "dq TRTEA SHIFTOFL" --show "dq TRTEC SHIFTOFL" --show "dq EMBA SHIFTOFL" --show "dq EMBC SHIFTOFL" --show "dq EMECA SHIFTOFL" --show "dq EMECC SHIFTOFL" --show "dq FCALA SHIFTOFL" --show "dq FCALC SHIFTOFL" --show "dq HECA SHIFTOFL" --show "dq HECC SHIFTOFL" --verbose --filenametag "data08*" --partition "ATLAS"

Selection sequence: Checking for runs in run range [[90270, 90350]] : 8 runs found
 Checking if number of events matches 100000+ : 8 runs found
 Checking if the filename tag matches "data08*" : 8 runs found
 Checking if partition name matches "ATLAS" : 8 runs found
 Checking in the DQ folder SHIFTOFLInfo in : png file
 data/atirunquery_h_Run_Events.png has been created

No. of runs selected: 8

Total no. of events: 7,099,530 (excluding 2 runs without available #events information)

Execution time: 2.0 sec

Run	Links	#LB	#Events	SCTB	SCTEA	SCTEC	TRTB	TRTEA	TRTEC	EMBA	EMBC	EMECA	EMECC	FCALA	FCALC	HECA	HECC
90270	RS, AMI, Trigger, ELOG	10	n.a.	G	G	G	U	U	U	G	Y	G	G	G	G	G	G
90272	RS, AMI, Trigger, ELOG	58	5,065,168	G	G	G	G	G	G	G	G	G	G	G	G	G	G
90275	RS, AMI, Trigger, ELOG	47	n.a.	G	G	G	U	U	U	G	Y	G	G	G	G	G	G
90295	RS, AMI, Trigger, ELOG	4	101,740	R	R	R	G	G	G	G	G	G	G	G	G	G	G
90300	RS, AMI, Trigger, ELOG	4	105,887	R	R	R	G	G	G	G	G	G	G	G	G	G	G
90311	RS, AMI, Trigger, ELOG	3	127,227	R	R	R	G	G	G	G	G	G	G	G	G	G	G
90329	RS, AMI, Trigger, ELOG	5	132,395	R	R	R	G	G	G	G	G	G	G	G	G	G	G
90345	RS, AMI, Trigger, ELOG	48	1,567,113	G	G	G	G	G	G	R	R	R	R	R	R	R	R

Summary:

8 runs			7,099,530														
--------	--	--	-----------	--	--	--	--	--	--	--	--	--	--	--	--	--	--

GoodRunsLists Tools & Usage

Package: DataQuality/GoodRunsLists, contains:

- Goodruns-list 'configurations', from which goodruns-lists are created.
- Root-based tools to interpret goodruns-lists, and to apply operations to them.
- Athena tools for selection of good runs/lumiblocks in Athena, based on good runs list or directly from COOL.

DQ group's responsibility to keep up-to-date & distribute all goodruns-lists & their configurations.

Interaction w/ Atlas Run Query

ATLAS Run Queries

Run Summaries Trigger Configuration Query AMI Data Search DDM Dashboard Tier-0 Monitoring DQ Monitoring Data Preparation Operations

Run Search – Insert Your Query:

f r 90270-90350 and dq em y+ and dq pixb y+ / sh dq pix,sct,em,til

Show Runs

Examples (query format inspired by SPIRES):

Run and event ranges	Time ranges and duration	Detectors	Streams	Magnets	Data quality	Project tag	Trigger	Partition
find run 90270-90350 and events 100000+ / show run and events								
f r 90270-90350 and ev 100k+ / sh r and ev								
f r 90270-90350 and ev 100k+ / sh r and ev and ftag and part / nodef								
f r 90270+ and ev 100k-								
f r 90270-90350,90500+								
f r 90270,90275,90380 and ev 100k-200k								

(More formatting help)

Search Result

Selection rule: f r 90270-90350 and dq em y+ and dq pixb y+ / sh dq pix,sct,em,til

Query command: AtlRunQuery.py --run "90270-90350" --dq "EMBA yellow+ SHIFTOFL" --dq "EMBC yellow+ SHIFTOFL" --dq "EMECA yellow+ SHIFTOFL" --dq "EMECC yellow+ SHIFTOFL" --dq "PIXB yellow+ SHIFTOFL" --show run --show events --show "dq PIXB SHIFTOFL" --show "dq PIX0 SHIFTOFL" --show "dq PIXEA SHIFTOFL" --show "dq PIXEC SHIFTOFL" --show "dq SCTB SHIFTOFL" --show "dq SCTEA SHIFTOFL" --show "dq SCTEC SHIFTOFL" --show "dq EMBA SHIFTOFL" --show "dq EMBC SHIFTOFL" --show "dq EMECA SHIFTOFL" --show "dq EMECC SHIFTOFL" --show "dq TILBA SHIFTOFL" --show "dq TILBC SHIFTOFL" --verbose --filename tag "data08" --partition "ATLAS"

Selection sequence:

Checking for runs in run range [[90270, 90350]]	: 8 runs found
Checking in the DQ folder SHIFTOFL (EMBA) and (EMBC) and (EMECA) and (EMECC) and (PIXB) is 'yellow+'	: 3 runs found
Checking if the filename tag matches "data08"	: 3 runs found
Checking if partition name matches "ATLAS"	: 3 runs found

No. of runs selected: 3

Total no. of events: 5,065,168 (excluding 2 runs without available #events information)

Execution time: 2.1 sec

■ <http://atlas-runquery.cern.ch/>

■ GoodRunsLists package interacts with AtlRunQuery.py tool to get good runs list.

■ Convert this query cmd-line into structured, transparent DQ flags configuration

Example GoodRunsList configuration

- Example command to make new dq flags selection:
> definenew_goodrunslis MyGoodRunsList

```
from GoodRunsLists.GoodRunsListConfig import GoodRunsListConfig
```

```
class MyGoodRunsList(GoodRunsListConfig):
```

```
    def __init__(self):  
        GoodRunsListConfig.__init__(self)
```

```
    ## Specify each selection of dq flags here.
```

```
    ## The query is interpreted as the logical AND of all elements in the array.
```

```
    ## All selection criteria at: Database/CoolRunQuery/python/AtlRunQueryParser.py
```

```
    ## See http://atlas-runquery.cern.ch for more user examples.
```

```
    dqflagsquery = [  
        'em    green LBSUMM',  
        'pixb green LBSUMM',  
        'tile green LBSUMM',  
    ]
```

```
    ## Set the dqflags query
```

```
    self.querydict['dq'] = dqflagsquery
```

```
    self.listname      = "MyGoodRunsList.xml"
```

- Some selections controlled in base class by DQ group.

- Transparent DQ flags selection you wish to be applied to lumi blocks.
- Normally, physics groups select on secondary DQ flags only.

End-user only concerned with DQ-flags selection.

Base GoodRunsList configuration

Selections hidden from end-user

```
class GoodRunsListConfig:
    def __init__(self):
        # -- run cmd options
        self.querytool          = "AtlRunQuery.py"
        self.listname           = "default.xml"

        # -- query selections in AtlRunQuery.py
        # -- see Database/CoolRunQuery/python/AtlRunQueryParser.py for full list
        self.querydict = {}
        self.querydict['partition'] = ["ATLAS"]
        self.querydict['ftag']      = ["data08*,data09*"]
        self.querydict['events']   = ["10000+"]
        self.querydict['run']      = ["90000+"]
        self.querydict['duration'] = ["120s+"]
        self.querydict['time']     = []
        self.querydict['detector'] = ['all']
        self.querydict['streams']  = []
        self.querydict['mag']      = ["solenoidon","toroidon"]
        self.querydict['dq']       = []
        self.querydict['smk']      = []
        self.querydict['trigger']  = []
        self.querydict['projectTag'] = []
        self.querydict['release']  = []
        self.querydict['conditions'] = []

        # -- DAQ partition
        # -- filename tag: collection filename prefix
        # -- number of events per run
        # -- run number selection range
        # -- duration of run selection
        # -- time of run selection, eg.
        # -- subdetectors turned on, eg.
        # -- trigger output streams, eg.
        # -- magnetic field configuration
        # -- selection of data quality flags
        # -- super master key, eg.
        # -- select specific triggers available, eg.
        # -- project tag, eg:
        # -- release used, eg: '15.1.*'
        # -- COOL conditions tag to use, eg: 'COMCOND-HLTC-001-00'
```

- List of all selection criteria accepted by AtlRunQuery tool.
 - Run#, # of events, time, magn.field, dq flags, etc.
- GoodRunsListConfig class serves as base selection class for other GRL configurations ... (see previous slide)
 - Eg, centrally update conditions tag or run-range of all GRLBs definitions.

Example GoodRunsList definition

- Example script to interact with AtlRunQuery:
 - > create_goodrunlist MyGoodRunsList
 - This produces the goodruns list: data/MyGoodRunsList.xml ...

```
[mback@lxplus236]~% create_goodrunlist MyGoodRunsList
>> Now processing good runlist definition: 'MyGoodRunsList'
>> Calling cmd equivalent of:
AtlRunQuery.py "find run 90000+ and partition ATLAS and detector all and mag solenoidon
and mag toroidon and duration 120s+ and ftag data08*,data09* and events 10000+ and dq em
y+ and dq pixb y+ and dq til y+"
Data source lookup using /afs/cern.ch/atlas/software/releases/15.1.0/DBRelease/6.8.1/XML
Config/dblookup.xml file
```

- Note selection AtlRunQuery selection query between "..."
- You can cross-check/study selection by plugging query back into:
 - <http://atlas-runquery.cern.ch/>

Example GoodRunsList creation

```
[mbaak@lxplus244]~% create_goodrunslis ExampleConfig
Running at cern. Okay.
>> Now processing good runslis definition: 'ExampleConfig'
>> Calling cmd equivalent of:
AtlRunQuery.py "find run 90000+ and partition ATLAS and detector all and mag solenoidon and mag t
roidon and duration 120s+ and ftag data08*,data09* and events 10000+ and dq em y and dq pixb g+"
Data source lookup using /afs/cern.ch/atlas/software/builds/AtlasCore/15.1.0/InstallArea/XML/Atla
sAuthentication/dblookup.xml file
SELOUT Checking for runs in run range [[90000, 1073741824]] ==> 1080 runs found
SELOUT Checking if duration of run is more than 120 seconds ==> 882 runs found
SELOUT Checking if number of events matches 10000+ ==> 463 runs found
SELOUT Checking if [detector mask & 72554882531319] matches 72554882531319 ==> 45 runs found
SELOUT Checking in the DQ folder SHIFTOFL
SELOUT (PIXB) is 'green+'
SELOUT (EMBA) and (EMBC) and (EMECA) and (EMECC) is 'yellow' ==> 10 runs found
SELOUT Checking if the filename tag matches "data08*,data09*" ==> 10 runs found
SELOUT Checking if partition name matches "ATLAS" ==> 10 runs found
SELOUT Checking if the ['SolCurrent'] is within [7700,+inf] ==> 10 runs found
SELOUT Checking if the ['TorCurrent'] is within [20000,+inf] ==> 10 runs found
-----
```

- GoodRunsList stored as: data/ExampleConfig.xml

GoodRunsList for TopMix sample

- **TopMix sample:**
 - Containing W's, Z's, top quarks, etc.
 - <https://twiki.cern.ch/twiki/bin/view/AtlasProtected/TopMixingExercise>
 - Dummy DQ flags exist in COOL for MC run 108175.
 - [user.RichardHawkings.0108175.topmix_Muon.AOD.v5](#)
- **Good-runs list can now also be made for MC TopMix sample!**
 - Thanks to Joerg Stelzer and Andreas Hoecker
- **Example TopMix good runs list configuration added to GoodRunsLists package**
 - `python/TopMixConfig.py`
- `create_goodrunlist TopMixConfig`
 - **Good-runs list stored as: `TopMixConfig.xml`**

Example GRL xml file

- Common run/LB xml format agreed-upon for good run list file.

- Desing by TAG grp

- Simple and transparent structure

- Ranges of LBs

- Also contains versioning and metadata info

- Name, version nr.
- Search query used to create xml file.
- Streams, trigger names of interest (work in progress)

```
<?xml version="1.0" ?>
<!DOCTYPE LumiRangeCollection
  SYSTEM 'http://atlas-runquery.cern.ch/LumiRangeCollection.dtd'>
<!--Document is created by AtlRunQuery.py-->
<LumiRangeCollection>
  <NamedLumiRange>
    <Name>My Good Run List</Name>
    <Version>2.1</Version>
    <Metadata Name="Query">
      find db MC and dq em green LBSUMM and
      dq pix green LBSUMM and dq tile green LBSUMM
      and dq sct green LBSUMM and run 108175
      and ctag DetStatusLBSumm-TopMix1
    </Metadata>
    <Metadata Name="Stream"/>
    <LumiBlockCollection>
      <Run>108175</Run>
      <LBRange Start="0" End="1"/>
      <LBRange Start="1351" End="1576"/>
      <LBRange Start="1775" End="28396"/>
    </LumiBlockCollection>
  </NamedLumiRange>
</LumiRangeCollection>
```

Tools to interpret goodrunlist(s)

- Root-based tools to interpret, operate, write good run lists (xml)
 - Can be used equally in Athena, ARA, pyroot, C++ standalone.
 - Supports standalone compilation:
 - `cd cmt; make -f Makefile.Standalone`

GoodRunsLists/run/example.py

```
## read the goodrunlist xml file(s)
reader = Root.TGoodRunsListReader('LumiRangeCollectionA.xml')
#reader.AddXMLFile('LumiRangeCollectionB.xml')
reader.Interpret()
goodrunlist = reader.GetMergedGRLCollection()

## show summary of content
goodrunlist.Summary() ## Detailed: Summary(True)

## constains run and/or lumiblock ?
if goodrunlist.HasRun (90210) : print "Foo!"      # contains run 90210 ?
if goodrunlist.HasRunLumiBlock (90210,7) : print "Bar!" # run 90210, lb 7 ?

## Now write out xml again ...
writer = Root.TGoodRunsListWriter(goodrunlist,'test2.xml')
writer.WriteXMLFile()
```


Operations to GoodRunsLists

- Boolean operations that can be applied to GoodRunsList objects:
 - AND, OR, !XOR
- See eg (root macro) : GoodRunsLists/run/example.C

```
// get sum (OR) of grl1 with grl2
Root::TGoodRunsList sumgrl      = grl1.GetSumWith(grl2);

// get overlap (AND) of grl1 with grl2
Root::TGoodRunsList overlap     = grl1.GetOverlapWith(grl2);

// get part only in grl1, not in grl2
Root::TGoodRunsList onlyingr1   = grl2.GetPartOnlyIn(grl1);

// get part not in grl1, only in grl2
Root::TGoodRunsList notingr1    = grl2.GetPartNotIn(grl1);
```

- Useful for adding new runs / lumiblocks !

Athena DQ flags Selector tool

```
# Configure the goodrunlist selector tool
from GoodRunsLists.GoodRunsListsConf import *
ToolSvc += GoodRunsListSelectorTool()
GoodRunsListSelectorTool.GoodRunsListVec = [ 'grl1.xml', 'grl2.xml', 'grl3.xml' ]
GoodRunsListSelectorTool.BlackRunsListVec = [ 'blacklist1.xml' ]

# add LumiBlockMetaDataTool, used in RecExCommon_topOptions.py
from LumiBlockComps.LumiBlockCompsConf import LumiBlockMetaDataTool
ToolSvc += LumiBlockMetaDataTool( "LumiBlockMetaDataTool" )

## default top sequence
from AthenaCommon.AlgSequence import AlgSequence, AthSequencer
job = AlgSequence()
job += AthSequencer("ModSequence1")

## GRL selector, which used GoodRunsListSelectorTool and TriggerDecisionTool
job.ModSequence1 += GRLTriggerSelectorAlg('GRLTriggerAlg1')
job.ModSequence1.GRLTriggerAlg1.GoodRunsListArray = ['grl1']
job.ModSequence1.GRLTriggerAlg1.BlackRunsListArray = ['blacklist1']
job.ModSequence1.GRLTriggerAlg1.TriggerSelection = 'EF_mu20'
```

- GRLSelectorTool accepts good-run and black-run xml files

Example event selection

- What you see in Athena (with verbose output):

```
AthenaEventLoop... DEBUG initialize()
--- <DEBUG> TGoodRunsListReader : Read xml data card: "TTbarGoodRunsList.xml"
--- <DEBUG> TGoodRunsListReader : XML parser returned code: 0
--- <DEBUG> TGoodRunsListReader : child node value: "TTbarSample"
--- <DEBUG> TGoodRunsListReader : child node value: 2.1
--- <DEBUG> TGoodRunsListReader : Metadata value: My query
--- <DEBUG> TGoodRunsListReader : Metadata: Name = "Query"
--- <DEBUG> TGoodRunsListReader : subchild node value: 105200
--- <DEBUG> TGoodRunsListReader : LBRange: Start = "0"
--- <DEBUG> TGoodRunsListReader : LBRange: End = "7"
```

```
AthenaEventLoop... DEBUG passEvent() :: Event accepted by goodruns list.
```

```
AthenaEventLoop... DEBUG passEvent() :: Event rejected by black list.
```

- Example in GoodRunsLists/run/ :
 - athena GoodRunsLists/GoodRunsLists_jobOptions.py

Two GRL/Trigger selections

- Two GRL/trigger selections, in separate sequences
 - Eg. Corresponding two different trigger selections

```
## the (default) top sequence
from AthenaCommon.AlgSequence import AlgSequence, AthSequencer
job = AlgSequence()
job += AthSequencer("ModSequence1")
job += AthSequencer("ModSequence2")

## Sequence1
job.ModSequence1 += GRLTriggerSelectorAlg('GRLTriggerAlg1')
job.ModSequence1.GRLTriggerAlg1.GoodRunsListArray = ['grl1']
job.ModSequence1.GRLTriggerAlg1.BlackRunsListArray = ['blacklist1']
job.ModSequence1.GRLTriggerAlg1.TriggerSelection = 'EF_mu20'
## others algs go here ...

## Sequence2
job.ModSequence2 += GRLTriggerSelectorAlg('GRLTriggerAlg2')
job.ModSequence2.GRLTriggerAlg2.GoodRunsListArray = ['grl2','grl3']
job.ModSequence2.GRLTriggerAlg2.TriggerSelection = 'EF_e20_loose'
## other algs go here ...
```

Luminosity block book-keeping

- LumiBlockMetaDataTool does book-keeping of processed lumi-blocks, and applies DQ selection to lumi-blocks.
- It produces one XML file containing list of good, processed runs and lumi-blocks.

```
ToolSvc.LumiBlockMetaDataTool          INFO  stop method: Event Loop Completed  
--- TGoodRunsListWriter : GoodRunsList stored as : luminosity_grl_108175.3472-108175.3533.xml
```

- XML file contains:
 - List of all processed lumi blocks, all info used in DQ and trigger selection.
 - (Same name as LumiBlockCollection stored in output MetaDataStore)

Example lumiblock XML file (one seq.)

- XML file contains:
 - List of all processed lumi blocks, all info used in DQ and trigger selection.
 - Same name as LumiBlockCollection in MetaDataStore

```
<?xml version="1.0"?>
<!DOCTYPE LumiRangeCollection SYSTEM "http://atlas-runquery.cern.ch/LumiRangeCollection.dtd">
<!--This document is created by GoodRunsListWriter. Ask Max Baak for details.-->
<LumiRangeCollection>
  <NamedLumiRange>
    <Name>LumiBlocks_GoodDQ0</Name>
    <Version>10</Version>
    <Metadata Name="BadRunsList0">blacklist1</Metadata>
    <Metadata Name="Description">luminosity_Sequence0</Metadata>
    <Metadata Name="GRLQuery">_OR_grl1_AND_NOT_blacklist1</Metadata>
    <Metadata Name="GoodRunsList0">grl1</Metadata>
    <Metadata Name="SelectionTrigger">EF_mu20</Metadata>
    <Metadata Name="TriggerName0">EF_mu20</Metadata>
    <LumiBlockCollection>
      <Run>108175</Run>
      <LBRange Start="3476" End="3488"/>
      <LBRange Start="3490" End="3502"/>
      <LBRange Start="3504" End="3510"/>
      <LBRange Start="3518" End="3518"/>
      <LBRange Start="3521" End="3533"/>
    </LumiBlockCollection>
  </NamedLumiRange>
</LumiRangeCollection>
```

Merging of xml-files / ntuples

- $O(1000)$ xml-files back from GRID jobs. Q: Now what? A: merge!
- `merge_goodrunslists <dirWithXMLorROOTfiles>`

```
% merge_goodrunslists -t tree ./
o) Found 3 input root files.
o) Now merging ... (this takes some time)
o) Found 3 independent good-runs lists. Now writing merged xml files.
--- TGoodRunsListWriter : GRL stored as : merged_0_grl_108175.3476-108175.3533.xml
--- TGoodRunsListWriter : GRL stored as : merged_1_grl_108175.3472-108175.3490.xml
--- TGoodRunsListWriter : GRL stored as : merged_2_grl_132350.0-132350.1000.xml
```

GoodRunsLists/scripts/

- Script finds all xml-files or -strings in root files, in given directory.
- Merges GRLs where possible. Distinguishes between selections.
- Spits out all merged, independent GRLs (eg. one per selection).
- Automatically filters out any double lumi-blocks between GRLs !
- Can use each output XML as input for total luminosity calculation.
 - Or as input GRL for new athena job.

GoodRunsLists Summary

- The primary storage medium of DQ status flag information is the conditions DB (COOL)
- Once detector subsystems and CP groups have finalized DQ flags assessment for data → tag and lock COOL DB
 - Lock expected before every major reprocessing.
- Once locked, Good-Runs-Lists for physics analyses to be centrally provided by Data Quality group.
 - Distributed on GRID, in form of XML files.
 - Pick up these files in your D3PD production jobs.
- CP groups & DQ group currently in process of setting of secondary DQ flags (= flags for physics objects)
- Physics groups create Good-Runs-Lists configuration files
 - Your responsibility to interact with your physics group about this!
 - Turned into actual Good-Runs-Lists by DQ group.

Conclusion

- Now you try it yourself!
- Enjoy the rest of the tutorial ;-)
- <https://twiki.cern.ch/twiki/bin/view/Atlas/GoodRunsListsTutorial>

Further reading material:

- GoodRunsLists information:
 - <https://twiki.cern.ch/twiki/bin/view/Atlas/GoodRunsLists>
- ATL-COM-GEN-2009-015 : 'Official' GoodRunLists document
- DQ presentation by Andreas Hoecker:
 - <http://indico.cern.ch/materialDisplay.py?contribId=2&materialId=slides&confId=54930>

LBNL Tutorial Q&A

- <https://twiki.cern.ch/twiki/bin/view/Atlas/GoodRunsListsTutorial>
- Can ask tutorial questions on public skype chat:
 - LBNL ATLAS Analysis Jamboree
- Join by opening link below in your browser.
Be sure to change: yourskypeusername
to your skype user name
- [http://www.skype.com/go/joinpublicchat?
skypename=yourskypeusername&topic=LBNL%20ATLAS
%20Analysis
%20Jamboree&blob=8FEiMrCDvAhHbagpP9VkoyHGr_rin0U-7GSG
dSpBzCA_LDL2sIePS9w3nKV5](http://www.skype.com/go/joinpublicchat?skypename=yourskypeusername&topic=LBNL%20ATLAS%20Analysis%20Jamboree&blob=8FEiMrCDvAhHbagpP9VkoyHGr_rin0U-7GSGdSpBzCA_LDL2sIePS9w3nKV5)