

# Table of Contents

<b>How To Run LX2 alignment on Grid.....</b>	<b>1</b>
Release setup receipe 15.3.1.....	1
How to find which release is available on grid sites:.....	1
setup ganga at CERN:.....	1
Ganga Submit script:.....	2

# How To Run LX2 alignment on Grid

## Release setup receipe 15.3.1

To run the LX2 code please setup release 15.3.1 This can be done at CERN using a script script like the following:

```
# CERN ATLAS kit
# if you need a particular DB release use the following: export DBRELEASE_OVERRIDE=7.2.1
cd /afs/cern.ch/atlas/software/releases/15.3.1/cmtsite/
source setup.sh -tag=15.3.1,opt,32
cd -
cd /afs/cern.ch/user/c/cortiana/athena/AtlasOffline-15.3.1/
export CMTPATH=`pwd`:${CMTPATH}
cd -
```

Then you should checkout and compile the following packages:

```
# use get_tag to see which tag to use for InDetRecExample:
# for 15.3.1;the following should be used: AtlasReconstruction;/InnerDetector/InDetExample/InDetR

cmt co -r InDetRecExample-01-17-51-03 InnerDetector/InDetExample/InDetRecExample
cmt co -r InDetLocalChi2AlignEvent-00-00-12 InnerDetector/InDetAlignEvent/InDetLocalChi2AlignEven
cmt co -r InDetLocalChi2AlignTools-00-00-70 InnerDetector/InDetAlignTools/InDetLocalChi2AlignTool
cmt co InnerDetector/InDetAlignAlgs/InDetLocalChi2AlignAlgs
```

```
cd InnerDetector/InDetExample/InDetRecExample/cmt && source setup.sh && gmake && cd -
cd InnerDetector/InDetAlignEvent/InDetLocalChi2AlignEvent/cmt && source setup.sh && gmake && cd -
cd InnerDetector/InDetAlignTools/InDetLocalChi2AlignTools/cmt && source setup.sh && gmake && cd -
cd InnerDetector/InDetAlignAlgs/InDetLocalChi2AlignAlgs/cmt && source setup.sh && gmake && cd -
```

## How to find which release is available on grid sites:

To check whether release is available on grid, the following site can be useful:

[https://atlas-install.roma1.infn.it/atlas\\_install/list.php?sitename=MPPMU](https://atlas-install.roma1.infn.it/atlas_install/list.php?sitename=MPPMU) (to check what is available at MPPMU)

[https://atlas-install.roma1.infn.it/atlas\\_install/list.php?rel=15.3.1](https://atlas-install.roma1.infn.it/atlas_install/list.php?rel=15.3.1) (to check the status of 15.3.1 release in various sites)

For more general information about the central release status:

<http://atlas-computing.web.cern.ch/atlas-computing/projects/releases/status/>

## setup ganga at CERN:

```
source /afs/cern.ch/project/gd/LCG-share/current/etc/profile.d/grid_env.sh
source /afs/cern.ch/atlas/offline/external/GRID/ddm/DQ2Clients/setup.sh
source /afs/cern.ch/sw/ganga/install/etc/setup-atlas.sh
export LFC_HOST=lfc-fzk.gridka.de
export LCG_CATALOG_TYPE=lfc
echo "getting grid token"
voms-proxy-init -voms atlas
# the following open a tokens to RZG:
# because gangadir has been setup as a soft link to the /afs/home area at RZG.
```

```
# where I had more space available
echo "getting RZG token for ganga dir"
klog -cell ipp-garching.mpg.de -principal gcortian
```

## Ganga Submit script:

In the `InDetRecExample` area untar the following file

```
/afs/cern.ch/user/c/cortiana/public/tar/LX2ganga15.3.1.tar.gz
```

then:

```
cd gangadir
ganga
execfile('LX2IterMasterScript.py')
```

`gangadir` contains all relevant `jOp` file to run a complete LX2 alignment iteration. You should modify `LX2IterMasterScript.py` to change `IterationDirectory` and `LocalAMGA` definition according to your setup.

The `LX2IterMasterScript.py` looks like the following:

```
import time, os
RUNLOCAL=False # change this to run locally
MERGE = False # change to True for executing LX2 merging
MyPWD = os.popen('pwd').read().strip()
# IterationLoop
#####
for IterationNumber in range(1,2): # just one iteration for starters
    print 'submitting LX2 alignmet job iteration=', IterationNumber
    # AlignmentLevel settings
    #####
    AlignmentLevel=1
    print 'Alignment Level = ', AlignmentLevel
    # Setup
    #####
    j=Job()
    jname="TestLX2Grid_15.3.1_91900all_Iter%02d"%(IterationNumber)
    j.name=jname
    j.application=Athena()
    j.application.atlas_release='15.3.1'
    # backend
    #####
    if RUNLOCAL :
        j.backend=Local()
        #j.backend.queue = 'atlasidali'
    else :
        j.backend=LCG()
        j.backend.requirements=AtlasLCGRequirements()
        #j.backend.requirements.sites= ['CERN']
        j.backend.requirements.cloud = 'DE'
        #j.backend.requirements.excluded_sites=['LRZ-LMU', 'HEPHY-UIBK']
        j.backend.requirements.cputime=1440
        j.backend.requirements.memory=1024
    # jOp and config files
    #####
    if IterationNumber==1 :
        os.system ('pool_insertFileToCatalog NominalAlignment.pool.root')
        j.inputsandbox=['./PoolFileCatalog.xml', './NominalAlignment.pool.root', './RealCosmicsLocalAlignment.pool.root']
    else :
        os.system ('pool_insertFileToCatalog AlignmentIter_%02d.pool.root'%(IterationNumber-1))
        AlignPoolFile= './AlignmentIter_%02d.pool.root'%(IterationNumber-1)
        j.inputsandbox=['./PoolFileCatalog.xml', AlignPoolFile, './RealCosmicsLocalChi2Alignment.pool.root']
    # modify jOp file to set the
```

setup ganga at CERN:

## RunLX2AlignmentOnGrid < Sandbox < TWiki

```
# wanted iteration and alignment level
#####
os.system ('sed -e s/"TOCHANGEVALUE"/%d/ -e s/"TOCHANGELEVEL"/%d/ jobOptions_cosmic_LX2.py >
os.system ('chmod 755 myLX2iterjOp.py')
j.application.option_file='./myLX2iterjOp.py'
j.application.max_events=50
#j.application.max_events=-1
# The prepare() call creates a tar file of your user_area and ships it to the Grid.
#####
j.application.prepare()
# job splitters
#####
if RUNLOCAL:
    pass
else:
    j.splitter=DQ2JobSplitter()
    j.splitter.numfiles = 1
# inputdata
#####
if RUNLOCAL:
    j.inputdata=ATLASLocalDataset()
    j.inputdata.get_dataset_from_list('localdatasetnamelist.txt')
    full_print(j.inputdata.names)
else:
    j.inputdata=DQ2Dataset()
    j.inputdata.dataset=['data08_cosmag.00091900.physics_IDCosmic.merge.DPD_IDCOMM.o4_r653_p2
                        #'data08_cosmag.00091891.physics_IDCosmic.merge.DPD_IDCOMM.o4_r653_p
                        ]
    j.inputdata.number_of_files = 1

j.outputdata=None
# test output sandbox
#####
file1="LX2Align_%02d.txt"%(IterationNumber)
file2="LX2Align_%02d.pool.root"%(IterationNumber)
file3="NTuple_LX2Align_%02d_Monitor.root"%(IterationNumber)
file4="NTuple_LX2Align_%02d_AlignResults.root"%(IterationNumber)
file5="monitoring.root"
j.outputsandbox= [file1,file2,file3,file4,file5]
# job submission
#####
j.submit()

# job status checking
#####
# if more iterations are forseen,
# need to wait for all subjobs to
# be completed
#####
jid=j.id
jst=j.status
print "Submitted job ", jid, jst
jst=jobs(jid).status
thisj=jobs(jid)
print "Submitted job ", jid, jst
WaitingTime = 0
while str(jst)!='completed':
    time.sleep(60)
    WaitingTime += 60
    total=0
    completed=0
    failed=0
    running =0
    submitted =0
    # check every 5 min the job status
    if (WaitingTime%300==0 or WaitingTime==60):
        for sj in thisj.subjobs:
```

## RunLX2AlignmentOnGrid < Sandbox < TWiki

```
total+=1
if sj.status=='completed':
    completed+=1
elif sj.status=='failed':
    failed+=1
elif sj.status=='running':
    running+=1
elif sj.status=='submitted':
    submitted+=1
FS = float(submitted)/(total)*100.
FR = float(running)/(total)*100.
FC = float(completed)/(total)*100.
FF = float(failed)/(total)*100.
print 'Waited ' + str(WaitingTime) + ' seconds...' + 'Job stat W = %s%% R = %s%% C
else :
    print 'Waited ' + str(WaitingTime) + ' seconds...'
sys.stdout.flush()
jst=jobs(jid).status
if jst=='failed': break
print jobs(jid).status

if jst=='failed':
    print 'Some parallel jobs were failing, stopping alignment script!'
    break

# Subjob result merging
#####
# run locally a LX2 merger + ntuple merger
#####
if MERGE:
    # specify where to store merging results
    #####
    IterationDirectory='/afs/cern.ch/user/c/cortiana/athena/AtlasOffline-15.3.1/InnerDetector
    SearchDir=''
    if RUNLOCAL:
        # LocalAMGA dir: must point the your gangadir/workspace/cortiana/LocalAMGA directory
        #####
        LocalAMGA='/afs/cern.ch/user/c/cortiana/gangadir/workspace/cortiana/LocalAMGA'
        SearchDir = LocalAMGA+'/%d/output/*%02d_AlignResults.root' %(jid,IterationNumber)
        ntuple = LocalAMGA+'/%d/output/*Monitor.root' %(jid)
    else :
        SearchDir = LocalAMGA+'/%d/*/output/*%02d_AlignResults.root' %(jid,IterationNumber)
        ntuple = LocalAMGA+'/%d/*/output/*Monitor.root' %(jid)

print 'executing merging'
execfile('RealCosmicsMergeScript.py')
os.chdir(MyPWD) # switch back to Masterscript pwd for next iteration

print 'merging ntuples'
os.system ('root -l -q \'merge_ntuples_all.C(\"%s\", %d, %d,\"%s\")\'') % (ntuple, Alignme
print 'merging ntuples DONE'
```

-- GiorgioCortiana - 2009-09-02

---

This topic: [Sandbox > RunLX2AlignmentOnGrid](#)  
Topic revision: r1 - 2009-09-02 - [GiorgioCortiana](#)



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.  
or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)