

# Table of Contents

<b>manTreeMaker instructions.....</b>	<b>1</b>
<b>manTreeMaker &amp; manTree Versions.....</b>	<b>2</b>
<b>Package Installation.....</b>	<b>3</b>
Compiling in Athena 16.6.5.5.1.....	3
Compiling in Athena.....	3
<b>Running Job on Grid.....</b>	<b>5</b>
<b>Run.....</b>	<b>6</b>
<b>Downloading Output.....</b>	<b>7</b>
<b>Merging Output.....</b>	<b>8</b>

# manTreeMaker instructions

# manTreeMaker & manTree Versions

Tested Release	manTreeMaker tag	manTree tag
17.0.5.6.1	trunk / manTreeMaker-00-10-05	manTree-00-10-09
17.0.4.2.1	trunk	manTree-00-10-01
16.6.5.5.1	manTreeMaker-00-09-07	manTree-00-09-08
TopPhys,16.0.3.8.1	manTreeMaker-00-07-01	manTree-00-07-07
TopPhys,16.0.3.3.3	manTreeMaker-00-07-00	manTree-00-07-07
TopPhys,15.6.13.3.2	manTreeMaker-00-06-09	manTree-00-06-10

# Package Installation

To make manTrees, you need to work in the athena environment, so first setup your favorite release & then checkout & compile manTree as follows. Note that in the following I assume you have \$SVNMAN set to svn+ssh://svn.cern.ch/repos/atlasgrp/Institutes/Manchester. **Note:** You can only use \$SVNMAN if you're using one of the linux8 machines at Manchester and your username is the same at Manchester and CERN. You can freely replace \$SVNMAN with:

```
svn+ssh://*cernusername*@svn.cern.ch/repos/atlasgrp/Institutes/Manchester
```

## Compiling in Athena 16.6.5.5.1

If you are not using 16.6.5.5.1, then this section can be skipped, otherwise you will first need to install the egammaAnalysisUtils, JetMomentTools & TruthUtils packages before following the 'Compiling in Athena' section:

```
cd $TestArea
cmt co -r egammaAnalysisUtils--00--02--18 Reconstruction/egamma/egammaAnalysis/egammaAnalysisUtils
cd Reconstruction/egamma/egammaAnalysis/egammaAnalysisUtils/cmt/
make
cd $TestArea
cmt co -r JetMomentTools-00-00-38 Reconstruction/Jet/JetMomentTools
cd Reconstruction/Jet/JetMomentTools/cmt
make
cd $TestArea
cmt co -r TruthUtils-00-00-08 Generators/GenAnalysisTools/TruthUtils
```

## Compiling in Athena

To compile in athena, first set up an athena release & then checkout the package & compile:

```
cd $TestArea
cmt co $SVNMAN/SFrame/manTree -d SFrame/manTree
cd SFrame/manTree/cmt
make -j
```

To checkout a tag you need to add `-r manTree-XX-YY-ZZ` to the `cmt` command.

Then checkout and compile the package needed for the making of the trees:

```
cd $TestArea
cmt co $SVNMAN/ManAthenaTools/manTreeMaker -d ManAthenaTools/manTreeMaker
cd ManAthenaTools/manTreeMaker/cmt
make -j 2
```

To checkout a tag you need to add `-r manTreeMaker-XX-YY-ZZ` to the `cmt` command

Also checkout the package with the grid submission scripts;

```
cd $TestArea
cmt co $SVNMAN/ManAthenaTools/ManGridUtils -d ManAthenaTools/ManGridUtils
cd ManAthenaTools/ManGridUtils/cmt
make
```

**Note:** You can only use \$SVNMAN or \$SVNOFF if you're using one of the linux8 machines at Manchester and your username is the same at Manchester and CERN. You can freely replace \$SVNMAN as above and \$SVNOFF with:

## ManTreeMaking < AtlasSandbox < TWiki

svn+ssh://\*cernusername\*@svn.cern.ch/repos/atlasoff/

To run the maker, with no selection cuts applied to the events, you can use the `treeMaker_jobOptions.py` file in the share directory. In this example we are using the file `manTreeMaker/run/topInputsTreeMake_NoFilt_RecExCommon_topOptions.py`, but you can use any `*Option.py` file of your choice.

# Running Job on Grid

As an example we will consider making manTree samples with no selection cuts. There are also similar files for electron and muon skims.

To submit a job with no cuts to the grid you need to use the file gridscripts/topinputsNuSubmit.py in manTreeMaker. This file looks like:

```
from ManGridUtils.submitScriptMake import submitScriptMake

run = "_071010_v1"
num_aods_per_subjob = 10
jo = "topInputsTreeMake_NoFilt_RecExCommon_topOptions.py"
outputfile = "tree.root"
excluded_sites = ['ANALY_QMUL']
mergeOnGrid = True
```

If we examine this:

- *run* is a tag to distinguish this job and will be put on the end of the dataset output name
- *num\_aods\_per\_subjob* specifies the number of aods per subjob. Due to space allowances this number should be small (~4) for MC files
- *jo* is the job option in the share or run directory you wish to use
- *mergeOnGrid* specifies whether or not to merge the output ntuples on the grid or not. It is recommended to set this to true

Looking at file input. In the gridscripts directory there are several \*list.py files. For electron data this is el\_aod\_list.py: This looks like

```
periodG1_container ={
    '165632' : 'data10_7TeV.00165632.physics_Egamma.merge.AOD.f293_m609',
    '165591' : 'data10_7TeV.00165591.physics_Egamma.merge.AOD.f292_m609'
}

periodG2_container ={
    '165732' : 'data10_7TeV.00165732.physics_Egamma.merge.AOD.f293_m609',
    '165703' : 'data10_7TeV.00165703.physics_Egamma.merge.AOD.f293_m609'
}
```

Looking back at the file gridscripts/topinputsNuSubmit.py, these lists are used to input files for your job:

```
#AOD files
from el_aod_list import periodG1_container as container
from el_aod_list import periodG2_container
container.update( periodG2_container )
```

Now look at topInputsTreeMake\_NoFilt\_RecExCommon\_topOptions.py. The AOD input is given by something like:

```
#The AOD input file
from AthenaCommon.AthenaCommonFlags import athenaCommonFlags
athenaCommonFlags.FilesInput = ['/var/tmp/joel_tmp/mc10_7TeV(...)/AOD.299419._000975.pool.root.1']
```

The *root.1* file in directory */var/tmp/joel\_tmp/mc10\_7TeV(...)/* will be made into a manTree. (**Hint:** if you save the \*root. file into a /tmp/ directory then remember which PC you did this on!)

# Run

Once you have set up Athena, set up dq2 and SFrame do:

```
cd $TestArea/ManAthenaTools/manTreeMaker/gridscripts/  
python topinputsNuSubmit.py  
mv runme.py ../run/
```

This will create a runme.py submission script. Open a **new** terminal then set up ganga as well as Athena and SFrame (but not dq2) and do:

```
$TestArea/ManAthenaTools/manTreeMaker/run/  
ganga runme.py
```

# Downloading Output

Use these instructions if you used the merging on the grid.

Once the grid job has finished you can merge the output using the ganga jobID In a new terminal setup dq2 & athena. Then do

```
cd $TestArea/ManAthenaTools/manTreeMaker/gridscripts/  
python  
from RetrieveMergePanda import doRetrieve  
jobs=[ID, ID2...]  
doRetrieve(jobs)
```



# Merging Output

Use these instructions if you did not use the merging on the grid.

Once the grid job has finished you can merge the output using the ganga jobID In a new terminal setup SFrame and dq2. Then do

```
cd $TestArea/ManAthenaTools/manTreeMaker/gridscripts/  
python  
from RetrieveMergePanda import doMerge  
jobs=[ID, ID2...]  
doMerge(jobs)
```

This will create a directory in /tmp/username/X/jobID/ with the Merged output and a text file for both Events Processed and Events Written.

**Note:** You may be asked to change \$DQ2\_LOCAL\_SITE\_ID to UKI-NORTHGRID-MAN-HEP\_LOCALGROUPDISK. If so then do:

```
export DQ2_LOCAL_SITE_ID=UKI-NORTHGRID-MAN-HEP_LOCALGROUPDISK
```

-- JohnAlmond - 28-Oct-2010 -- MarkOwen - 25-Oct-2011 -- SteveMarsden - 31-Oct-2011

---

This topic: AtlasSandbox > ManTreeMaking  
Topic revision: r17 - 2012-01-11 - MarkOwen



Copyright &© 2008-2022 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