Fixing ROOT Problem with gfortran libs (on Yosemite)

After installing gfortan from http://hpc.sourceforge.net/, I could no longer use pyROOT - python would crash as soon as I tried to import ROOT. The fix turned out to be:

- `sudo port selfupdate`
- `sudo port upgrade outdated`
- recompile ROOT from scratch

Reco Transform for exclusive samples

On lxplus do:

```
setupATLAS
asetup 17.0.5.6,slc5
localSetupPandaClient
pathena "--trf=Reco_trf.py inputAODFile=%IN outputNTUP_SMWZFile=%OUT.NTUP_SMWZ.root" "--inDS=mc11
```

(replace "ahamil" with your username, and IN / OUT datasets with appropriate datasets)

Feynman diagrams in Latexit on OSX 10.8

So simple...

- install from Texshop Download Page
- follow step 2 of Taku's page (I didn't need to do step 1)
- there is no "Composition" panel in Latexit 2.5, the relevant stuff is under the "Typesetting" panel

ROOT v5.34 on ubuntu 11.04 with python 2.7 (PGLabs machines)

Installing ROOT v5.34 on the PGLab machines:

- `sudo apt-get install gcc g++ make binutils libx11-dev libxpm-dev libxft-dev libxext-dev python-dev`
- download source from ROOT
- unzip tarball, move `root` folder to `/home/user/Applications/root_v5.34.05`
- `./configure linux    --prefix=/usr/local/ --with-x11-libdir=/usr/lib/i386-linux-gnu --with-xft-libdir=/usr/lib/i386-linux-gnu --with-xext-libdir=/usr/lib/i386-linux-gnu`
- `make`
- `make install`
- `sudo ldconfig`
- add `source /home/user/Applications/root_v5.34.05/bin/thisroot.sh` to `.bashrc`

ROOT v5.34 on mac osx 10.6.8 with python 2.7

Upgrading ROOT to v5.34, I did the following:

- download source from ROOT
- unzip tarball in `/Applications/root\_v5.34` folder (moved content of `root` folder to `root\_v5.34`)
- `./configure macosx --enable-python --enable-builtin-freetype` (freetype thing might not have been necessary, I put that in when `macosx64` flag was failing)

- `make -j2`
- add the following to my `.bash\_profile` (last two necessary because of non-default python location)
    - `source /Applications/root_v5.34/bin/thisroot.sh`
    - `export PYTHONDIR=/Library/Frameworks/Python.framework/Versions/Current`
    - `export PATH=$PYTHONDIR/bin:$PATH`

Panada Setup

Panda instructions on Panda Twiki (see "Information for Users").

I use `prun` to run panda without Athena.

For LXPLUS, essentially just need to:

- `export PATHENA_GRID_SETUP_SH=/afs/cern.ch/project/gd/LCG-share/current_3.2/etc/profile.d/grid_env`
- `source /afs/cern.ch/atlas/offline/external/GRID/DA/panda-client/latest/etc/panda/panda_setup.sh`
- I put these in my bash profile as alias `psetup`

This is an example to execute a completely general job. Here is a python script, call it `purepython.py`

```
import sys
print sys.argv
f = open('out.dat','w')
f.write('hello')
f.close()
sys.exit(0)
```

Then run it with

```
$ prun --exec "python purepython.py" --outDS user.ahamil.test1234
```

Go to Panda Monitoring page to see how your job is doing.

- Under "Quick Search" on the left you can enter your "PandaID" number that was printed when you submitted the job.
- You also get an email when your job is finished.

When it's finished you can download the output dataset using dq2, for example `dq2-get user.ahamil.test1234` will download the job output.

Printing to Minolta printer with a MAC

Just in case any one else in the department has a mac:

- Download the driver from: http://solutions.konicaminolta.com.au/support/DriverPublic.cfm (select bizhub 363)
    - download should be a file called 423_Series_MacOS_v1.3.0.sit
- Open the 423_Series_MacOS_v1.3.0.sit file, should create a directory with name 423_Series_MacOS_v1.3.0
- In the directory, go to "Driver" -> "OS_10_Y_X" -> "A4" -> "bizhub_423_10Y.pkg" (Y is the particular version of OSX you have)
- Run the "bizhub_423_10Y.pkg" installer
- In you mac System Perferences, add the printer using the IP address and the Line Printer Daemon protocol (IP address 137.158.92.77)

    - select paper source PC-409, Finisher FS-529, Puncher None
- Open a document you want to print in Preview
    - Print the document
    - Select the printer you just installed
    - In the dropdown that says "Preview", select "Output Method"
        - select "Account Track" and enter you username in "Department" and your password in "Password"
    - In "Presets" dropdown, select "Save As" to save the Account Track settings for next time.

PyROOT on Mac OS 10.6.X

Instead of recompiling ROOT, I decided to install compatible versions: ROOT version 5.26 and Python 2.5.4 of the libraries (there may be other combinations of libraries that work, but I don't know what they are or where to find them)

- download the Python v2.5.4 from http://www.python.org/download/releases/2.5.4/ and install it following the instructions on the site
- download the ROOT v5.26 from http://root.cern.ch/drupal/content/production-version-526 and install it following the instructions on the site and in the ROOT instruction manual (found on the ROOT site)
- set some environment variables (in your `.bash_profile`) to get python to be able to read the ROOT libraries:
    - `export ROOTSYS=/Applications/root`
    - `export PATH=$ROOTSYS/bin:$PATH`
    - `export DYLD_LIBRARY_PATH=$ROOTSYS/lib:$DYLD_LIBRARY_PATH`
    - `export PYTHONDIR=/Library/Frameworks/Python.framework/Versions/Current`
    - `export PATH=$PYTHONDIR/bin:$PATH`
    - `export DYLD_LIBRARY_PATH=$PYTHONDIR:$DYLD_LIBRARY_PATH` * `export PYTHONPATH=$ROOTSYS/lib:$PYTHONPATH`

You might find this pystartup file useful to make your python do tab completion and history.

Installing ATHENA via Pacman kit

Official ATLAS instructions are at:
https://twiki.cern.ch/twiki/bin/viewauth/Atlas/WorkBookInstallingAtlasSoftware

Here is what I did on my 64bit SLC5 machine.

Get and install pacman:

- `wget` http://atlas.bu.edu/~youssef/pacman/sample_cache/tarballs/pacman-latest.tar.gz
- `mkdir /usr/local/atlas` (if it doesn't already exist)
- `mv pacman-latest.tar.gz /usr/local/athena/`
- `cd /usr/local/atlas`
- `tar -zxf pacman-latest.tar.gz`
- `cd pacman-*`
- `source setup.sh`
- `cd /usr/local/atlas`

Install an Athena release (17.0.6 in this example):

- `cd /usr/local/atlas/pacman-*`
- `source setup.sh`
- `mkdir /usr/local/atlas/athena`
- `mkdir /usr/local/atlas/athena/17.0.6`

- `cd /usr/local/atlas/athena/17.0.6`
- `pacman -get am-CERN:17.0.6`

I can't seem to get the Kit Validation `pacman -get am-CERN:17.0.6+KV` to work, it fails in trying to access some server in rome... Note that you can also get the pacman command for a specific release, at http://atlas-computing.web.cern.ch/atlas-computing/projects/releases/status/ (click on the release and scroll to the bottom of the page)

Now add the following to your `.bashrc` file:

- export path_to_kit=/usr/local/atlas/athena/17.0.6
- export AtlasSetup=/$path_to_kit/AtlasSetup
- alias asetup='source $AtlasSetup/scripts/asetup.sh'

When you start a new terminal you should now be able to do:

- `asetup 17.0.6`
- `athena -i`

There was a problem compiling the PhysicsAnalysis, so I also needed to install some 32-bit package (I'm not sure why):

- `yum install glibc-devel-2.5-65.el5_7.1.i386`

VNC via SSH tunnel

If you have a machine behind a firewall that you would like to VNC to, but do not have direct ssh access to that machine, you can ssh tunnel to the machine if you have some server behind the firewall that you do have access to. For example: logging into the geneva cluster through lxplus, or logging into your UCT desktop via phy.uct.ac.za.

Some nomenclature:

- `XXX` is your local machine, ie. your laptop
- `YYY` is the machine inside the firewall that you have access to (ie. lxplus.cern.ch or www.phy.uct.ac.za)
- `ZZZ` is the machine you want to access (ie. geneva cluster or your desktop machine)

In other words, you are working on `XXX` and you will tunnel through `YYY` to get to `ZZZ`

The instructions:

- Login to ZZZ to start a VNC server session:

```
ssh YYY
ssh ZZZ
vncserver
```

Note, this session stays alive until it is explicitly killed (see last point). When starting the vncserver for the very first time (you're not having a ~/.vnc direcoty yet), you will be prompted for a password. This is the password you'll have to use when connecting later on (vnc passwords can be set using the vncpasswd command).

The `vncserver` command gives you a session name, something like

```
New 'ZZZ.uct.ac.za:1 USERNAME' desktop is ZZZ.uct.ac.za:1
```

This tells you that the vncserver is listening on ZZZ.ucta.ac.za:1 (this is the SESSION_NAME which you need for killing it later).

- From XXX machine create an SSH tunnel to YYY

```
ssh -g -f -L 5901:localhost:5901 -x -e none YYY sleep 200000
```

The message 'bind: Address already in use' can be ignored.

- Login to the YYY machine to create a 2nd tunnel to ZZZ:

```
ssh YYY
ssh -g -f -L 5902:localhost:5901 -x -e none ZZZ sleep 200000
```

The 1 in ':5901' corresponds to display 1. This has to be the number returned by the vncserver!

* From XXX start the vncviewer:

```
vncviewer localhost:2
```

the ':2' specifies to use display 2, the one that we forwarded to display 1 on the ZZZ machine. Use the password you gave when you first started the vnc server.

- Killing the VNC server session, on the ZZZ machine do:

```
vncserver -kill MY_SESSION_NAME
```

VNC connection to lxplus

Connect to lxplus using Chicken of the VNC from my mac

Start the VNC server on lxplus:

- `ssh lxplus.cern.ch vncserver`

Returns something like: `New 'lxplus433.cern.ch:1 (ahamil)' desktop is lxplus433.cern.ch:1` This means"=lxplus433.cern.ch:1=" means it is running on lxplus433's desktop 1

Create an ssh tunnel to your machine (`localhost`):

- `ssh lxplus433.cern.ch -L5801:localhost:5801 -L5901:localhost:5901`

(replace the '1' in 5801 and 5901 with the desktop number and the '433' with the values returned above)

Start Chicken of the VNC on your mac:

- put in Host: `localhost` and Display: `1`

The password is set when you first start the vnc server on lxplus (or by using the `vncpasswd` command)

installing ROOT on Ubuntu 9X pglab03 machine

```
wget ftp://root.cern.ch/root/root_v5.30.04.source.tar.gz  ##get ROOT, pick your favourite version
tar -xzvpf root_v5.30.04.source.tar.gz
cd root
```

```
WHERE_TO_INSTALL_ROOT=/usr/local      ## feel free to change this

./configure --with-x11-libdir=/usr/lib --with-xft-libdir=/usr/lib --with-xext-libdir=/usr/lib --w

make
sudo make install                      ## sudo needed at least for for /etc/root

cd $WHERE_TO_INSTALL_ROOT
source bin/thisroot.sh
root                                   ##  :-)   works...
```

Then add this to your .bash_profile

```
source /usr/local/bin/thisroot.sh
export LD_LIBRARY_PATH=$ROOTSYS/lib:$PYTHONDIR/lib:$LD_LIBRARY_PATH
export PYTHONPATH=$ROOTSYS/lib:$PYTHONPATH
```

mounting afs on OSX 10.6

Download the binary from http://www.openafs.org/macos.html, install it (use "cern.ch" as the "cell name"). You should now see `/afs/` mounted on your local drive. Use `klog` to authenticate with kerberos so that you can see your afs area.

compiling LHAPDF on OSX 10.6

Installed LHAPDF 5.8.1 on OSX 10.6:

- get LHAPDF from http://projects.hepforge.org/lhapdf/
- followed install instructions, got the error listed at http://projects.hepforge.org/lhapdf/usertips/MacOs_problems
- used fink (http://www.finkproject.org/) to install gcc45 (command `fink install gcc45`)
    - fink binaries are put in `/sw/bin/` and are named like `gcc-4` so they don't conflict with existing compilers on you computer
- configured with `./configure --prefix=/your/install/directory --enable-low-memory CC=/sw/bin/gcc-4 CXX=/sw/bin/g++-4 FC=/sw/bin/gfortran` (as suggested in Mac Os problems page)
- ran `make`, got another error (`error: command 'gcc' failed with exit status 1`) after `Making all in pyext`
- re-configured with above configure command plus `--disable-pyext`
- `make` and `make install` completed successfully!

renewing Grid Certificate

Official instructions: https://twiki.cern.ch/twiki/bin/view/AtlasComputing/WorkBookStartingGrid

Below are my (potentially outdated) instructions:

When you get the mail saying you need to "renew or request a new" User Certificate:

- go to https://ca.cern.ch
- click "Request or renew user certificate using Mozilla browser"
- follow instructions to request a new certificate
- go back to https://ca.cern.ch and revoke the old certificate

Now you need to move the certificate to where ever you want to use the grid:

- go to Firefox (4.0.1) -> Preferences -> Advanced -> View Certificates
- find the new certificate in the list, and click "Backup"
    - you will be requested to make up a password
    - a blah.p12 file will be made
- copy the blah.p12 file to lxplus (or where ever)
- do the following

```
> openssl pkcs12 -in mycert.pfx -clcerts -nokeys -out usercert.pem
> openssl pkcs12 -in mycert.pfx -nocerts -out userkey.pem
> chmod 400 userkey.pem
> chmod 444 usercert.pem
```

- you will be asked for another password (this is your 'grid' password)
- then move these two .pem files to the `.globus` directory (if you don't have one, make one!)

You should now be able to do:

```
> source /afs/cern.ch/atlas/offline/external/GRID/ddm/DQ2Clients/setup.sh
> voms-proxy-init -voms atlas'
```

problems with and feynmp

When Latexit gets "stuck", try to delete the contents of the temp folder: `/var/folders/sW/sWtXsIPxEOqc8ab3fkHk8++++TI/-Tmp-/LaTeXiT-2.3.0/`

You can find the location of the temp folder in the logfile (do "Latex->Display Last Log")

hacking an eps file

To move text around in an eps file, open in a text editor, find the lines that look like:

```
gsave  2268 1948 0 0 C 1855 680  t 0 r /Helvetica findfont 53.2685 sf 0 0 m (ATLAS) show
NC gr
```

In this example

- text is "ATLAS"
- x and y positions of the text are 1855 and 680
- font type is Helvetica

landscape table in Latex

To put a Latex table in landscape, use the "turnpage" environment:

```
\begin{turnpage}
table blah...
\end{turnpage}
```

move files based on their creation time

```
for file in `ls`; do if [ `stat -f %m $file` -gt 1280304400 ]; then mv $file newDir/; fi;
done
```

list files without extension

This can be used to list files without their extension

```
ls -1 | sed 's/\(.*\)\..*/\1/'
```

If you want to rename all *.eps file or use them in a `convert` or `mv` command, you can use something like:

```
for file in `ls -1 *.eps | sed 's/.eps//'`; do convert $file.eps $file.pdf; done
```

join pdf files

To join two (or more) pdf files, there is an automator script written in mac os 10.4 and above:

```
ln -s /System/Library/Automator/Combine\ PDF\ Pages.action/Contents/Resources/join.py
joinPDF.py
```

Then you can use it to join, say input1.pdf and input2.pdf into a file final.pdf like this:

```
./joinPDF.py -o final.pdf input1.pdf input2.pdf
```

mac screen-capture

- apple-shift-4 is the shortcut for a screen-shot
- you can change the default file type by doing: `defaults write com.apple.screencapture type png` then log-out, log-in

tkdiff and SVN

By doing a "tkdiff local_file_name" you get a diff between the SVN base and you local file!

ROOT Setup

Put the following in a file called `.rootrc` in your afs home area:

```
# Path used to find macros.
Unix.*.Root.MacroPath:      .:/afs/cern.ch/user/a/ahamil/rootmacros:$(ROOTSYS)/macros:

# Rint (interactive ROOT executable) specific alias, logon and logoff macros.
Rint.Load:               /afs/cern.ch/user/a/ahamil/rootmacros/rootalias.C
Rint.Logon:              /afs/cern.ch/user/a/ahamil/rootmacros/rootlogon.C
Rint.Logoff:             /afs/cern.ch/user/a/ahamil/rootmacros/rootlogoff.C
```

Then you can put macros in your rootalias.C file, things like a start TBrowser shortcut (you only need to type "tb()" on ROOT prompt):

```
// start TBrowser________________________________________________________________________
void tb(int sx=550, int sy=400) {
  static int nb = 0;
  nb++;
  new TBrowser(Form("tb_%i",nb),Form("tb_%i",nb),sx,sy);
}
```

And you can put your default styles in rootlogon.C:

```
{
   gStyle->SetOptStat(11111111); // print overflows/underflows in the stat box
   gStyle->SetOptFit(1110); // print fit results in the stat box
   gStyle->SetPalette(1,0);
   gStyle->SetCanvasColor(10);
   gStyle->SetCanvasBorderMode(0);
   gStyle->SetFrameLineWidth(3);
   gStyle->SetFrameFillColor(10);
   gStyle->SetPadColor(10);
   gROOT->ForceStyle();
}
```

MacBook Setup

Here is what I did to setup my MacBook:

- initial software updates and preferences setup (no need to document...)
- get the following software from CERN (https://dfs.cern.ch/dfs/Departments/IT/Services/MacSupport/)
    - MS Office 2008, iWork '09, iLife '09, Mathmatica 7 (license server in readme file),
    - to get the licenses for these, go to http://it-div.web.cern.ch/it-div/gencomputing/mac-support/Licenses.htm and follow instructions to install K2Client
- I'm trying mac's Terminal which comes by default (seems ok so far)
- X11 is installed by default
- install "Aquamacs" xemacs, downloaded from http://aquamacs.org/
- install Firefox (icon in "stacks" should appear properly after re-boot)
- install MacTex (http://www.uoregon.edu/~koch/texshop/obtaining.html) which includes TexShop, LatexIt, convert, and many other tex extensions
- install "Chicken of the VNC" (linked to sourceforge from apple.com)
- install fink and fink commander from http://finkcommander.sourceforge.net/
- install XCode 3.1.3 Developer Tools from http://connect.apple.com (also available on install disk)
- install Fortran 77 compiler (g77) from http://hpc.sourceforge.net/ (g77-intel-bin.tar.gz)

CMT Version of a Package

Do `cmt show versions Trigger/TrigEvent/TrigEventAthenaPool` will show you both the version for the current release and the one you have checked out.

I've got a script which just checks out the correct version of a package for the release you have set up. It is available at `/afs/cern.ch/user/a/ahamil/scripts/checkout.sh`, to run it you need to `source checkout.sh`.

There is a script that does this already! It is `pkgco.py`.

Remove New Lines and Tabs in Emacs

New Lines: do a search and replace, replacing "C-q C-j" with " "

Tabs: replace "C-q " with " "

Trigger Validation Test

```
trigtest.pl --test AthenaModernRDO --rundir AthenaModernRDO --config
$AtlasArea/../../AtlasTrigger/$AtlasVersion/Trigger/TrigValidation/TriggerTest/Testing/trigValida
```

EvtMax in jobOptions

If you do an `EvtMax = 10` and yet still get the whole data file processed, try `theApp.EvtMax = 10`

Feynman Diagrams in LaTeXiT

To do Feynman diagrams in LaTeXiT:

- install "feynmp" from the FinkCommander
- follow the instructions at http://osksn2.hep.sci.osaka-u.ac.jp/~taku/osx/feynmp_latexit.html
- if you put the files in ~/texmf/, you need to run `mktexlsr` to tell latex where the files are
- my diagrams are saved in my LaTeXiT library

Who has SVN tag rights

Look at: http://atlas-computing.web.cern.ch/atlas-computing/projects/sit/atlasoff_authz.html

Tab completion in emacs

`ALT+/` is "tab completion" in emacs

Enhanced Biased Datasets

You can find the top and enhanced bias datasets defined here:

http://atlas-sw.cern.ch/cgi-bin/viewcvs-atlas.cgi/offline/Trigger/TrigValidation/TrigP1Test/test/TrigP1Test_TestConfig

cmthome setup

To allow a more flexible work area setup, you could change your ATLAS_TEST_AREA macro in your requirements file to:

```
macro ATLAS_TEST_AREA  "$PWD"
```

Then you

```
cd
source ~/cmthome/setup.sh -tag...
```

and it uses the work area where you are.

Dataset Numbers in AMI

With the new naming convention the configuration tag

e323_s400_d99_r474

really is critically important. All that the mc08 means is that it is Monte Carlo generated in 2008, according to the naming convention. If you want to find out what is in your MC sample (and you should) you always have to look this up (use eg the "Interpret config tag" box on the "Dataset Search" in AMI).

For this config tag, AMI clearly says for e323 that it is an FDR2 evgen process, that the simu+digi (s400, d99) was done in 13.0.40.3 and the reco (r474) is done with 14.1.0.4. It works! Please learn to use it!

DEBUG in One Algorithm

```
from AthenaCommon.AlgSequence import AlgSequence
topSequence = AlgSequence()

from TrigEgammaHypo.TrigEgammaHypoConf import TrigL2PhotonFex
TrigL2PhotonFex = TrigL2PhotonFex(OutputLevel=DEBUG)
topSequence += TrigL2PhotonFex
```

Tag Collection in TRIG nightly

- power user
- AtlasTrigger
- 14.X.0-TRIG browse
- OverRideLockAndUpdatePV

Reading Trigger from AOD

```
root [1] gSystem->Load("libCintex.so");
root [2] Cintex::Enable();
root [3] gSystem->Load("libTrigEventTPCnv.so");
root [4] TBrowser b; // now I'm able to access all variables
```

Getting Commit & Tag Rights

To get commit and tag rights to a package, email `atlas.release at cern.ch`

Setting up a pcache

```
rel=rel_6
source ./cmthome/setup.sh -tag=pcache,32,${rel},noTest
TestArea=`pwd`
CMTPATH=${TestArea}:$CMTPATH
source /afs/cern.ch/atlas/software/builds/nightlies/pcache/AtlasProduction/${rel}/AtlasProduction
```

New Grid Certificate

I let my grid certificate expire, here is what I did to get a new one:

- start from http://www.usatlas.bnl.gov/twiki/bin/view/Support/HowToAtlasVO in Firefox
-

Using Hephaestus to find memory leaks

See UsingHephaestus. It's built into athena so you basically just do:

```
athena --leak-check-execute myTopOptions.py >& leak.log
```

You may need to change the `jobproperties.perMonFlags.doMonitoring` to `False` in your jobOptions file, then you should get the leak check output in `LeakCheck.txt` (otherwise it might be in `ntuple.pmon.heph.log`)

Using gdb with athena

Running all of athena in debug mode takes too much time and memory. To compile just one package in debug, but to leave the rest of ATLAS in opt mode (taken from UsingDebugBuiltPackagesWithOptBuild):

Add the following lines to the requirements files of the package:

```
private
macro cppdebugflags '$(cppdebugflags_s)'
macro_remove componentshr_linkopts "-Wl,-s"
```

(the last line is probably only needed if it's a component library)

Now do

```
cmt config
make binclean
make
```

You should now have a library with debugging symbols (you can check with `nm`), suitable for use with `gdb` (see GdbDebugger).

To run athena with gdb, you can do `athena.py -d jobOptions.py`.

Using Valgrind

See UsingValgrind

In short:

- build debug libraries as above
- `source /afs/cern.ch/sw/lcg/external/valgrind/3.2.3/slc4_amd64_gcc34/_SPI/start.sh`
- `valgrind --leak-check=yes --trace-children=yes --num-callers=8 --show-reachable=yes `which athena.py` jobOptions.py >& valgrind.log`

Replace text in file

To replace all instances of a word with another word in a file(s):
`rpl 'oldtext' 'newtext' ./*`
(the binary is `/afs/cern.ch/user/a/ahamil/scripts/rpl`)

Remove files loop

```
for file in `cat rmlist.txt`; do rm $file; done
for file in `cat rmlist.txt`; do cvs remove $file; done
```

Database issues

If job fails with message like:

```
CORAL/RelationalPlugins/oracle    Error ORA-02391: exceeded simultaneous SESSIONS_PER_USER limit
CORAL/Services/ConnectionService  Warning Failure while attempting to start a session on connecti
RalSessionMgr     Info Instantiate a R/W RalSessionMgr for 'COOL_INDET/OFLP200'
RalSessionMgr     Info Connect to the database server
```

```
IOVDbSvc                                          ERROR dbConnection is not correctly initializ
```

one can use at cern a local replica by running beforehand

```
 setupLocalDBReplica_CERN.sh
```

Converting case

```
echo "ff1c2c40-cc13-4b1a-b805-e9271be8c9d6" | tr 'a-z' 'A-Z'
```

gives `FF1C2C40-CC13-4B1A-B805-E9271BE8C9D6`

LCG Computing Elements

To get a list of the LCG computing elements do: `lcg-infosites --vo atlas ce`

DB errors

If you get Persistency database errors, try to make the following links in your `run` directory:

```
ln -s /afs/cern.ch/user/a/atlcond/coolrep/sqlite130 sqlite130
ln -s /afs/cern.ch/atlas/project/database/DBREL/packaging/DBRelease/current/geomDB geomDB
```

Setting up nightlies

4 Apr 2007

To setup the nightlies, make a cmthome with requirements containing:

```
set   CMTSITE  CERN
set   SITEROOT /afs/cern.ch
macro ATLAS_DIST_AREA ${SITEROOT}/atlas/software/dist
macro ATLAS_TEST_AREA /afs/cern.ch/user/a/ahamil/path/to/testarea
use AtlasLogin AtlasLogin-* $(ATLAS_DIST_AREA)
```

Do the cmt stuff:

```
source /afs/cern.ch/sw/contrib/CMT/v1r20p20070208/mgr/setup.sh
cmt config
```

Set up a particular release:

```
source setup.sh -tag=rel_1,oneTag
source
/afs/cern.ch/atlas/software/builds/nightlies/dev/AtlasOffline/rel_1/AtlasOfflineRunTime/cmt/setup
```

This is for the "AtlasOffline" project in "rel_1". You can do it for other projects, like "AtlasTrigger", and for other releases, like rel_2, etc...

Set read permissions on afs

4 Apr 2007

```
fs setacl -dir blah/ -acl system:anyuser read
```

to do it recursively in `[dir]` do: `afind [dir] -t d -e "fs setacl -dir {} -acl [ACLS to set]"`

ATHENA can't find a file

3 Apr 2007

Try adding projects to the following environment variables: `DATAPATH`, `PYTHONPATH`, `JOBSEARCHPATH`

remove .xxx extension from files

5 Mar 2007

```
for oldfile in *.float; do newfile=${oldfile%.*float}; echo $oldfile $newfile; done
```

decoding compiler symbols

28 Feb 2007

When the compiler gives you something like: `undefined symbol: _ZN14TrigTauClusterC1Edddl`
you can decode the symbol by doing: `echo _ZN14TrigTauClusterC1Edddl  | c++filt`
which gives you something like: `TrigTauCluster::TrigTauCluster(double, double, double, long)`

getting a tag

26 Feb 2007

To figure out which tag of a package corresponds to a particular release do

- `get_tag package_name` (for example: `get_tag Trigger/TriggerRelease`)

This gives you a list with the tag corresponding to each ATHENA release (note that 12.x.x is 2.x.x)

/include problem

2 Feb 2007

If you get a compiler error like: `mkdir: cannot create directory `/include': Permission denied`
check to make sure that the `$TestArea` environment variable is set correctly!

Command to make hists...

19 Dec 2006

=for var in $(less varlist.txt); do echo $var" = new TH1F(\""$var"\",\";"$var";\",100,0, 1);"; echo "sc = add_histogram("$var");" ; echo 'if(sc.isSuccess()) return StatusCode::FAILURE;' ; echo " "; done=

Get Rid of Collection Tree in aan

29 Nov 2006

To get rid of the Collection Tree in the Athena Aware Ntuple (in 11.0.5), I commented out line 482 `m_tree->Fill();` of the file `PhysicsAnalysis/AnalysisCommon/AnalysisTools/AnalysisTools-00-00-24/src/AANtupleStream.cxx`

Getting a Grid Certificate

18 Sep 2006

I started with: https://twiki.cern.ch/twiki/bin/view/Atlas/LJSFGettingStarted
Which lead me to: http://service-grid-ca.web.cern.ch/service-grid-ca/help/user_req.html
On `lxplus` I executed `~gridca/scripts/cert-req` from my home directory.
The output of the command is at =lxplus:~/.globus/grid_cert.log=
Armin.Nairz@cernNOSPAMPLEASE.ch is the guy who checks the certification request.
He pointed me to this page to follow the rest of the steps:
http://lcg.web.cern.ch/LCG/users/registration/registration.html (I'm now done step 2)
Got an email from my CERN CA with my certificate, which I copied to `lxplus:~/.globus/usercert.pem` (and made it readable to only me)
After importing the certificate into my browser I went to
https://lcg-voms.cern.ch:8443/vo/atlas/vomrs?path=/RootNode&action=execute to "register".
I think I am now an ATLAS VO "applicant"... whatever that means...

Debugging

14 Sep 2006

`athena XXX_jobOptions.py -i` runs athena in interactive mode
`athena XXX_jobOptions.py -b` runs athena in batch mode
`athena XXX_jobOptions.py -s` outputs where it is in the jobOptions file as it executes
`athena XXX_jobOptions.py -d` runs athena in debug mode with =gdb=
Quick Lessons about GDB:

- `sharedlib` will make `where` put out the file and line number instead of just the .so file
- `up` and `down` will move up and down in the stack
- `list` gives you the lines of code surrounding the present location

nohup

13 Sep 2006

Use `nohup` for remote running. For example:

```
nohup athena.py jobOptions.py > athena.log &
```

CASTOR

13 Sep 2006

Regular unix commands don't work with CASTOR. Use the following:

- `nsls /castor/cern.ch/user/a/ahamil/` to list
- `rfcp` to copy files
- `rfmkdir` to make directories

Margins in pdf

13 Jul 2006

White margins appear in converting ppt to pdf.
To get rid of them: Adobe Acrobat -> Tools -> Print Production -> Crop Pages -> Remove White Marginsg

This topic: Main > MiscComputing
Topic revision: r108 - 2015-06-17 - AndrewHamilton