



AAF tips and tricks

Arsen Hayrapetyan

Yerevan Physics Institute, Armenia

Software packages: aliroot, root, geant, alien

- Should I download the source and compile it?
 - You can, of course, but you can also download precompiled binaries. (See <http://aaf.cern.ch/node/81>)
 - Benefits:
 - Exactly the same packages used on AAF and grid
 - Much less time consuming
 - Correct dependencies for aliroot
 - Easy to use (you just source a shell script)
 - You can have multiple installations and switch between them
 - Available for Linux (SLC, Ubuntu...)
 - Mac support is underway

Enabling aliroot on AAF (I)

- You can get the list of packages available on AF with *gProof->ShowPackages()*
- Once you have selected the aliroot package, you can enable it on AF with *gProof->EnablePackage()*
- By default, only the “base” aliroot libraries are loaded (*libVMC*, *libTree*, *libPhysics*, *libMatrix*, *libMinuit*, *libXMLParser*, *libGui*, *libSTEER*, *libESD*, *libAOD*, *libANALYSIS*, *libANALYSISalice*)
- You can load different subsets of aliroot libraries for different needs: reconstruction, simulation, etc.
 - Different subsets are loaded through different “aliroot modes”
 - ALIROOT
 - REC
 - SIM

Enabling aliroot on AAF (2)

- You can also load extra libraries, e.g. CORRFW and PWG2resonances. Here is an example:

- `TList * list = new TList;`
- `list->Add(new TNamed("ALIROOT_MODE", "ALIROOT"));`
- `list->Add(new TNamed("ALIROOT_EXTRA_LIBS",
"ANALYSIS:ANALYSISalice:CORRFW:PWG2resonances"));`
- `TProof::Open("alice-caf.cern.ch");`
- `gProof->EnablePackage("VO_ALICE@AliRoot::v4-21-05-AN", list);`

...

05:16:46 16613 Mst-0 | Info in <lxbsql409.cern.ch>: Running in ALIROOT mode ...

05:16:46 16613 Mst-0 | Info in <lxbsql409.cern.ch>: Loading libANALYSIS.so ...

05:16:46 16613 Mst-0 | Info in <lxbsql409.cern.ch>: Loading libANALYSISalice.so ...

05:16:46 16613 Mst-0 | Info in <lxbsql409.cern.ch>: Loading libCORRFW.so ...

05:16:46 16613 Mst-0 | Info in <lxbsql409.cern.ch>: Loading libPWG2resonances.so ...

05:16:46 16613 Mst-0 | Info in <lxbsql409.cern.ch>: Enabling VO_ALICE@AliRoot::v4-21-05-AN was OK ...

Enabling aliroot on AAF (3)

- You can enable your own libraries using PAR files. Here is an example of enabling own TOF library:
 - `TList * list = new TList;`
 - `list->Add(new TNamed("ALIROOT_MODE", "ALIROOT"));`
 - `list->Add(new TNamed("ALIROOT_EXTRA_LIBS", "ANALYSIS:ANALYSISalice"));`
 - `list->Add(new TNamed("ALIROOT_EXTRA_INCLUDES", "TOF"));`
 - `gProof->EnablePackage("VO_ALICE@AliRoot::v4-21-05-AN", list);`
 - `gProof->UploadPackage("MyTOFPar.par");`
 - `gProof->EnablePackage("MyTOFPar");`

Enabling aliroot on AAF using the plugin

- Here is an example of loading aliroot libraries + extra CORRFW and PWG2resonances libraries:

```
#include "AliAnalysisAlien.h"
```

```
AliAnalysisAlien* plugin = new AliAnalysisAlien();
```

```
plugin->SetRunMode("full");
```

```
plugin->SetAliRootMode("ALIROOT");
```

```
plugin->SetAliROOTVersion("v4-21-05-AN");
```

```
plugin->SetAdditionalLibs("libCORRFW.so libPWG2resonances.so");
```

```
....
```




Common user problems

Can't connect after TProof::Reset()

- Sometimes (e.g. when you kill your proof session with Ctrl + C) you need to reset the connection. Resetting means killing your user proof processes on cluster machines. This requires some time.
 - **Remedy:** Wait for ~1 min before reconnecting. This pause should be enough for proof to clean up your previous session completely.

My task crashes and I see “bad_alloc” error

- There are memory limits per worker on AAF (3GB on resident and 3GB on virtual memory). When you hit them (no memory can be allocated) you might see the following message: “*terminate called after throwing an instance of 'std::bad_alloc'*”.
- **Remedy:** Check your code for memory leaks.

Authentication problems (I)

- You are getting the following error:

```
100525 09:32:02 001 Proofx-E: Conn::Authenticate: cannot obtain credentials (protocol: gsi)
XrdSec: No authentication protocols are available.
100525 09:32:02 001 Proofx-E: Conn::Authenticate: unable to get protocol object.
100525 09:32:02 001 Proofx-E: Conn::GetAccessToSrv: client could not login at [lxbsq1412.cern.ch:1093]
100525 09:32:02 001 Proofx-E: Conn::Connect: failure: cannot obtain credentials for protocol: Secgsi: ErrParseBuffer: error getting user proxies: kXGS_init
100525 09:32:02 001 Proofx-E: XrdProofConn: XrdProofConn: severe error occurred while opening a connection to server [lxbsq1412.cern.ch:1093]
```

Remedy: Do one of the following:

- Invoke `gEnv->SetValue("XSec.GSI.DelegProxy", "2");` before connecting to the cluster.
- Add the following line to your user `.rootrc` file:
`XSec.GSI.DelegProxy 2`

Authentication problems (2)

- You are getting the following error:

```
root [0] TProof::Open("canao@alice-caf.cern.ch")
100824 12:21:42 001 Proofx-E: Conn::Authenticate: cannot obtain credentials (protocol: gsi)
XrdSec: No authentication protocols are available.
100824 12:21:42 001 Proofx-E: Conn::Authenticate: unable to get protocol object.
100824 12:21:42 001 Proofx-E: Conn::GetAccessToSrv: client could not login at [lxbsq1409.cern.ch:1093]
00824 12:21:42 001 Proofx-E: Conn::Connect: failure: cannot obtain credentialsfor protocol: Secgsi: ErrParseBuffer: unknown CA: cannot verify server certificate: kXGS_init
100824 12:21:42 001 Proofx-E: XrdProofConn: XrdProofConn: severe error occurredwhile opening a connection to server [lxbsq1409.cern.ch:1093]
(class TProof*)0x0
```

Remedy: You should tell proof where are your trusted CA certificates. This can be done setting environment variable `X509_CERT_DIR` before connecting to AF:
`export X509_CERT_DIR=$ALIEN_ROOT/globus/share/certificates`

Management of the output in “proof” mode

- Quite often the problems are caused by the violation of any of the following rules:
 - When running in “proof” mode with plug-in or directly running analysis on AAF, you should NOT delete the output in the destructor:
 - AliAnalysisTask::~~AliAnalysisTask () {delete fOutput;} **is wrong!**
 - Correct version of the destructor is:
 - AliAnalysisTask::~~AliAnalysisTask() {
if(fOutput && ! AliAnalysisManager::GetAnalysisManager()->IsProofMode()) delete fOutput; }
 - The owner should be set in UserCreateOutputObjects():
 - fOutput->SetOwner(kTRUE);
 - At the end of UserCreateOutputObjects() one should have:
 - PostData(1, fOutput);
 - All data members of your task class must be initialised.

I want a dataset to be staged...

- You can request dataset staging from AAF team, posting a staging request in AAF savannah:
<https://savannah.cern.ch/bugs/?func=additem&group=aaf>
 - Following details must be specified:
 - Dataset name
 - Your e-mail (Originator email, determined automatically when you log in)
 - Cluster(s) on which the dataset should be staged
 - The dataset will be staged under /alice/data or /alice/sim
- You can stage the dataset yourself following instructions at <http://aaf.cern.ch/node/160>. (staging macro)
 - The dataset is staged under <your group>/<your username>, e.g. /PWG3/zampolli
- We are planning to create a page on AAF where you can request the staging and mark it “official” (i.e. to be staged under /alice/data or /alice/sim). The dataset with redirector paths will be then created and copied to alien and staging daemons on different clusters will “pick it up” later and stage it on every cluster.

My dataset is not fully staged...

- For example:

- gProof->ShowDataSets("/PWG2/jgrosseo/LHC10e13_121040")

| Dataset URI | # Files | Default tree | # Events | Disk | Staged |
|--------------------------------|---------|--------------|-----------|------|--------|
| /PWG2/jgrosseo/LHC10e13_121040 | 1576 | /esdTree | 5.708e+05 | 17GB | 90% |

- If you want your dataset to be staged completely, you should send an e-mail to Martin or me, and we'll complete/repair it for you.
- In the future a macro will be available for users to complete the staging. (like one for staging a dataset)

Links

- AAF user manual: <http://aaf.cern.ch/node/89>
- AAF FAQ: <http://aaf.cern.ch/node/57>
- AAF user support: <http://aaf.cern.ch/node/90>



**Thank you!
and...**

Questions?