

## AMORE setup

# 1. Installation

Follow these instructions : [Howto configure alice daq yum repository](#).

Type (with root privileges, the two first lines are needed to remove the native mysql)

```
rpm -e mysql-libs mysql mysql-devel postfix
rm -rf /var/lib/mysql/
yum install amore amore-web MySQL-server MySQL-shared MySQL-client MySQL-devel ACT date
```

The package ACT is not mandatory unless you develop amoreQA.

The package date is mandatory for development, but not if you just run AMORE.

After installation the following directory structure exists:

/opt/amore

|--amore-config : link to bin/amore-config

|--bin : all binaries and scripts

|--cmake : CMake find scripts

|--include

| |--amore : all AMORE headers

|--lib : all AMORE libraries

|--web : all web application of AMORE

### Important note about ROOT

Many people already have ROOT or prefer to compile it themselves. If it is the case, make sure you configured it with the options

```
--enable-soversion
--enable-mysql
```

In any case, the installation of AMORE will force the installation of ROOT in /opt !

To decide which version of ROOT to use, yours or the new one, update accordingly your environment variables.

# 2. Configuration

## 2.1 Environment setup

The following variables have to be set. We advice to add these lines to the .bashrc

```
# DATE
if [ -f /date/setup.sh ]; then
    . /date/setup.sh
```

```

fi
# AMORE and DIM
export AMORE=/opt/amore
export AMORE_SITE=/amoreSite
export LD_LIBRARY_PATH=${AMORE_SITE}/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH=${AMORE}/lib:$LD_LIBRARY_PATH
export PATH=${AMORE}/bin:$PATH
export LD_LIBRARY_PATH=/opt/dim/linux:$LD_LIBRARY_PATH
# AMORE extras
export DATE_RUN_TYPE=PHYSICS
export DATE_RUN_NUMBER=1
export AMORE_CDB_URI=local:///local/cdb
export DIM_DNS_NODE=localhost

```

If you didn't compile and install ROOT yourself, add

```

# ROOT
export ROOTSYS=/opt/root
export PATH=$ROOTSYS/bin:$PATH
export LD_LIBRARY_PATH=$ROOTSYS/lib:$LD_LIBRARY_PATH

```

Finally, add the commands

```

if [ -f /date/setup.sh ]; then
    export DATE_SITE=/dateSite
    . /date/setup.sh
    /date/runControl/start_dim.sh
fi

```

to the file /etc/rc.local. This way, DIM will be always started at boot time.

Before continuing, please close the terminal and open a new one.

## 2.2 Database configuration

1. If mysql was just installed now, you have to set a root password. To do so
  - a. Note down the password written in /root/.mysql\_secret
  - b. /sbin/service mysql start
  - c. mysqladmin -u root -h localhost -p password
  - d. Enter the password as in the file, then the new one + confirm
2. To setup the MySQL database and accounts to be used by AMORE, execute amoreMySQLSetup and follow the instructions.
3. Then, execute newAmoreSite to setup the necessary files in \$AMORE\_SITE.
4. Finally, please execute the following command and put it in your .bashrc :
 

```
. amoreSetup $AMORE_SITE/AMORE.params
```

## 2.3 Troubleshooting

*Problem* : The MySQL server runs on your local machine and, after entering your local machine name in the MySQL server host name field, you get the following error :

ERROR 1130 (HY000): Host 'yourhostname' is not allowed to connect to this MySQL server  
Connection failed.

*Solution* : Enter 'localhost' in the MySql server host name field.

## 3. Test run of the system

The Developer's kit (also called amoreTST) is a simple yet fully operational AMORE module. We will use it to test our system (see below). You can then use it as a skeleton to start developing your own module. It shows many usages and features of the framework. You can also simply use it to play with the system.

If you plan to develop your own AMORE module(s), follow the instructions in the section [Download and compilation of amoreTST](#), for those who just want to run AMORE, go directly to the section [Installation of amoreTST via YUM](#). In both cases, continue with the instructions in [Test run](#) and in case of problems head to the [Troubleshooting](#) section.

### 3.1 Download and compilation of amoreTST

1. Either download the tarball from the AMORE website or check it out from SVN:  
svn co [https://svn.cern.ch/repos/alicedaq/Software/amore\\_modules/branches/curren...](https://svn.cern.ch/repos/alicedaq/Software/amore_modules/branches/curren...)
2. Inside the top directory run make. This will create the shared libraries for this module.
3. Then execute make install. It installs the libraries in \$AMORE\_SITE/lib
4. If no build errors are encountered you should be able to see 3 shared libraries inside \$AMORE\_SITE/lib.

### 3.2 Installation of amoreTST via YUM

Type: `yum install amoreTST`

### 3.3 Test run

You must have followed the instructions of the two previous chapters for these instructions to work. We will now check whether your system has been correctly set up by starting the agent and the client user interface of the TST module (developer's kit).

1. Agent creation: First, create an agent that will use the module amoreTST. Please execute newAmoreAgent and follow the instructions. Simply choose the default values proposed by the script.
2. Data source preparation: To run, we need a data source. In production it will be a GDC or an LDC, but here we will use a raw data file. Retrieve a root file from AliRoot and use the `deroot` utility from AliRoot to transform it to a raw data file. If the file date from Run 1, use in addition `/opt/date/monitoring/Linux/mpConvert` on it.  
If your lab is equipped with a complete DAQ setup, you can of course later use your GDCs or LDCs as online data source (please refer to the DATE manual to know the proper syntax). Please note that `deroot` is part of AliRoot and is built only if DATE is available when building aliroot.  
You can also contact an AMORE developer to ask for a test file.

3. Agent launching: To start the agent, issue the command:

```
amoreAgent -a TST01 -s -r
```

is the path to the raw data file prepared at step 2. The datasource needs to provide data for a sufficiently long period. In case the file is too short, one can use the `-r` option to automatically rewind the file. Every 10 seconds (default monitor cycle length) the agent will publish updated objects to the pool, making them available to the clients. There are many more command line options that can be specified. For a complete and up-to-date list, run `amoreAgent` without options.

4. GUI startup: To start the GUI, in another terminal issue the command:

```
amore -d TST -m MyModule1
```

TST is the detector code and MyModule1 is the module name, e.g. the name of the UI class. A GUI window should open. Press the Start button. If the `amoreAgent` named `amoreAgentTST01`, started with the instructions of step 2, is running properly, you should be able to see some plots updated every 10 seconds.

5. Optionally, you can use the generic gui. Install it (`yum install dqmv`) and run it (`amoreGui`).

### 3.4 Troubleshooting

1. I get errors about "inserting agent info into the logbook". What should I do ?

Nothing, it is a feature that can work only at P2, i.e. in production. Just ignore it.

2. I get the error "Server Connecting to DIM\_DNS on localhost: Connection Refused", what can I do ?

The DIM DNS server is not running or the environment variable `DIM_DNS_NODE` is not pointing to the correct machine. To start the server do `"/date/runControl/start_dim.sh"`. To be noted that this command should be part of `rc.local` as mentioned above.

`DIM_DNS_NODE` should point to the machine and have access to the machine where the server is running.

3. `amoreAgent` complains that "Data file not found or not accessible" or "data corrupted". What can I do ?

It might be that you mistyped the path to the file. It might as well be that you are trying to read a raw data file containing events with the CDH v2 (Run 1) with a DATE monitoring library > 7.52 expecting CDH v3. If so, simply convert the raw data file using the utility `mpConvert` available with DATE.

4. The client can't connect, it says that connection has timed out. What is the problem ?

When this happened in the past, it was due to the machine was not able to ping itself by its hostname because of a wrong `/etc/hosts`. Make sure you can ping yourself with you hostname and public ip and that others can as well.

## 4. Further reading

If you wish to understand more about AMORE, read the [part 1 of the manual](#).

If you want to develop your module right away, jump to [part 2 of the manual](#) and to the [API description](#).

Finally, if you want to configure further your test and development setup, go directly to [part 3 of the manual](#).

## Annex A: Installation of AMORE from source

This is not the recommended way of installing AMORE. You still need to install the dependencies, probably with `yum` (see above). Also, the configuration part for AMORE's dependencies must be already done (see next section).

The source code is in SVN (see [this quick guide](#) attached to this page), in :

<https://svn.cern.ch/repos/alicedaq/Software/amore>

[https://svn.cern.ch/repos/alicedaq/Software/amore\\_modules](https://svn.cern.ch/repos/alicedaq/Software/amore_modules)

To build and install AMORE from source code (we use the trunk here), do:

1. `svn co https://svn.cern.ch/repos/alicedaq/Software/amore/trunk amore`
2. `cd amore`
3. `mkdir build`
4. `cd build`
5. [install cmake with yum if you don't have it]
6. `cmake ..`
7. `make -j8`
8. `make install`

And then for the web tools :

1. `cd ../src/web`
2. `mkdir dirty ; cd dirty ; cmake ..`
3. `make install`

The steps above will install amore in /opt. In order to customize the installation directory or the compilation options, add the configure options in points 5 and 2 above, with:

`cmake configure_options`

for example: `cmake -DCMAKE_INSTALL_PREFIX=destination_dir`

Instructions to use cmake can be found [here](#) (amongst other resources).

#### Attached files:

 [svnTuto.pdf](#)

 [svnTuto.odt](#)

---

**Source URL:** <https://alice-daq.web.cern.ch/products/amore-setup>