APE and Offline installation

APE

The Auger Package Environment will install and build the following packages for use with Offline: **boost**, **cmake**, **clhep**, **xerces**, **fftw**, **root**, **fdeventlib**, **cdas**, **cppunit**, **aires**, **geant4**. If you have any of these packages already installed, it may be wise to remove them or comment out their paths in the .bashrc prior to completing this installation, as it could create compatibility issues.

Installation packages

Before starting to download ape, some packages are prerequired. For installation you may use

apt-get install <name of package>

Some packages are not provided by ape but are recommended to be installed in addition:

- libncurses-dev (required for ccmake)
- libbz2-dev (required for boost)
- zlib1g-dev (required for boost)

It's also worthwhile to install the root pre-requirements.

To ensure that you have the newest applications, please run

apt-get update && apt-get upgrade

Downloading source

Download the latest version using

git clone git://devel.auger.unam.mx/ape.git

Note| Wherever you choose to install, you may be required to use sudo in front of every command that is listed thereafter.

Note| Whenever you are asked for a user and password, use the Auger ones

Installation

You do not need to configure or build APE; you can simply use the executable.

To adjust the installation directory and speed up the process, a configuration file can be used (~/.aperc).

The content of this file can look like the following:

```
[DEFAULT]
base = %(home)s/auger/software/ApeInstalled

[ape]
mirrors = mx
jobs = 4
```

The number of jobs can be set to the number of cores (+1).
After adjusting it, run

<path to ape executable>/ape install externals

where the <path to ape executable> is the path to ape-version/ape.

For details on this, see the APE User Guide.

Installation takes around 2-3h (depending on the number of parallel jobs) and requires an initial internet connection. Afterwards, to verify installation of the packages, execute the following from the directory where you executed the ./ape install externals command:

./<path to ape executable>/ape installed

Creating pointers

To create pointers to your recently installed APE files, you will need to add

eval './ape sh externals'

to your .bashrc.

Offline

The Auger Subversions Repository contains the recent versions of all programs/files of Auger. In particular, it has the newest developed version of Offline located in auger/Offline/trunk/.
To copy this directory of the SVN to your computer, use

svn checkout https://devel-ik.fzk.de/svn/auger/Offline/branches/v3r3

Afterwards (and from then on), you can update your file version by using svn update within the directory you have the repository (ex. ../../trunk).
For reference, here is the Auger SVN usage page, which contains details of proper usage and development of the SVN.

Alternately, you can use

git svn clone https://devel-ik.fzk.de/svn/auger/Offline/branches/v3r3

Installation

(a) Select a directory where you would like to build Offline, let's call it $build_dir.
mkdir $build_dir
cd $build_dir
(b) As detailed at the Offline CMake Notes Wiki, type
cmake $src_dir -Dprefix=$inst_dir

Where
$src_dir is the the directory of your source files (trunk)
$inst_dir is the desired path for installation, preferably not the same as the build directory since you want to be able to delete the build directory after successful installation. The install directory will be created automatically, no need to create.

If you receive errors, make sure that your system is pointing to the correct libraries as mentioned before.

(c) Inside your build directory, type make (and count on ~40 minutes or more of waiting). You can also use make -j4 if you wish to parallelize the process. Should the build fail, however, the output you see in your terminal may not be related to the fail.

(d) Type make install. This will automatically create $inst_dir and copy only the relevant files from the build to the install directory. After successful installation you are allowed to remove the build directory.

(e) Depending on your original directories, your installation could occur in a number of places. Regardless, to set the Offline environment, find the auger-offline-config executable (i.e. should be somewhere in the installation directory you specified) and add eval `<directory path>/auger-offline-config --env-sh` to your .bashrc.

Testing Offline

(a) Find the directory where the **Tutorials** folder is located. If this folder and the accompanying folders (e.g. SampleShowers, UserSkeleton) are located in /usr/..., copy them to your home directory.

(b) Enter the **StandardApplications/SdSimulation** folder. Type make clean and make. If your output culminates in an error free message, then your installation was successful. Otherwise, make sure you have followed all the instructions above, particularly as they relate to setting the environment (make sure to source the .bashrc).

Pyik

- Go to the directory where you want to place the pyik modules and do the following checkout

svn co https://devel-ik.fzk.de/svn/misc/PythonTools/pyik/

- Add the location of your pyik installation to your PYTHONPATH (in your .bashrc or whatever bash file you use to load the Offline environment)

PYTHONPATH=$PYTHONPATH:"*path to your pyik installation*"
export PYTHONPATH