

HAP Workshop Starting Tutorial

- HD cluster account
 - Make sure you have an account to the Heidelberg cluster
 - Make sure you have access to `/ifs/lf7/hess/users/username` or similar
 - When you get your account you will have access to `/home/username/`, for external people, however, this only gives you very limited storage space, which is not enough for running a production.
 - You need to check if besides that you also have a folder in the `hess` folder in `ifs`. It will very likely be located at `/ifs/l7/hess/users/username`.
 - If you don't have access to it, you need to request. (Back when I started, I had to talk to Lars about it, not sure how it is right now).
 - Anyway, if this is not ready for the workshop, we can still make it work, as long as you have at least a working cluster account.
- Setting up an ssh-key
 - For both HAP and `hdconfiggeneration` we will clone repositories from gitlab (<https://gitlab.in2p3.fr/hess/>). For that, it is useful to have an SSH Key.
 - You can check if you already have one at https://gitlab.in2p3.fr/-/user_settings/ssh_keys
 - If you don't, please follow an online tutorial which will be much better than my instructions ;) It should be quite easy to find one;
- Having your own local HAP installation
 - Clone the hap-chain-master repository

```
git clone git@gitlab.in2p3.fr:hess/software/hap-chain-master.git
```
 - Create a source file with the correct environment variables.
Create a `hap-source.sh` file with the following content:

```

module load cfitsio/4.3.0

source /ifs/l7/hess/software/root/root_v5.34.38/bin/thisroot.sh

export HESSROOT=~hess/hap-chain-master

export HESSDATA=/ifs/l7/hess/users/marandon/CalibData/CT5/hddata
_NewScheme2/
export HESSDST=/ifs/l7/hess/users/marandon/CalibData/CT5/dst_New
Scheme2/
echo "Your HESSDATA path is now $HESSDATA"
echo "Your HESSDST path is now $HESSDST"

export HESSVERSION="version36"
export HESSCONFIG="/ifs/l7/hess/analysis/$HESSVERSION"
echo "Your HESSCONFIG path is now $HESSCONFIG"

if [ -e ${HESSUSER}/scons/local/scons.py ] || [ -e ${HESSROOT}/scon
s/local/scons.py ]; then
    alias scons="scons.py"
    export PATH=$HESSROOT/scons/local/:$PATH
else
    unalias scons > /dev/null 2>&1
fi

export LD_LIBRARY_PATH=${HESSROOT}/lib:${LD_LIBRARY_PATH}
export PATH=${PATH}:${HESSROOT}/bin:${HESSROOT}/hddst/bin

unset SCONSFLAGS
export SCONSFLAGS=" --loglevel=warning fits=yes --c++11 --cc-flags
=-Wno-non-template-friend,-Wno-misleading-indentation"

echo "Your HESS env variables have successfully been set"

```

Source it:

```
source hap-source.sh
```

Change to the main folder of the repository:

```
cd hap-chain-master
```

Compile it:

```
scons HAP -j4
```

You should start getting A LOT of text rolling here. Be patient, take a coffee and hope for the best.

If it ends with `scons: done building targets.`, you should be good to go!

You can run a test analysis just to check if it worked:

```
$HESSROOT/hddst/bin/hap
```

Which should show you something like this, followed by all the HAP input options:

```
hap> -----  
hap> This is HAP version hap-dev-pl00.  
hap> This version of HAP is a development release.  
hap> It is neither foreseen nor recommended for analysis.  
hap> Please check for the latest stable release at the HAP wiki page.  
hap> -----
```

- hdconfiggeneration local installation

This is the main script for training the configurations, generating lookups and getting IRFs

- Clone the repository

```
git clone git@gitlab.in2p3.fr:hess/softwaremodules/hap/hapauxiliary/hdc  
nfiggeneration.git
```

- Make a branch for yourself (we likely will be implementing new things here)

```
git checkout -b workshop-yourname
```

- (Easy solution) instead of creating your own mamba env (which you can you can directly use mine or Tim's. For that add this to your `hap-source.sh` :

```
# >>> mamba initialize >>>  
# !! Contents within this block are managed by 'micromamba shell init' !!  
export MAMBA_EXE='/ifs/l7/hess/users/timu/software/micromamba/bin/mi  
cromamba';  
export MAMBA_ROOT_PREFIX='/ifs/l7/hess/users/timu/software/microma  
mamba';  
__mamba_setup="$("$MAMBA_EXE" shell hook --shell bash --root-prefix  
"$MAMBA_ROOT_PREFIX" 2> /dev/null)"  
if [ $? -eq 0 ]; then  
    eval "$__mamba_setup"  
else  
    alias micromamba="$MAMBA_EXE" # Fallback on help from micromam  
ba activate  
fi  
unset __mamba_setup  
# <<< mamba initialize <<<
```

You should be good to go. But anyway, things with the `hdconfiggeneration` are easier to be solved during the workshop.