SHSPE - SHow SPEctra

root macro to SHow SPEctra

1.1 download shspe from git

this is a standard procedure, (make and) go to some directory and then:

```
git init
git remote add origin https://github.com/jaromrax/shspe
git pull origin master
```

2.2 prerequisites for compilation of root:

basically, better check root.cern.ch website.

```
aptitude install dpkg-cross
aptitude install libxext-dev
aptitude install libxpm-dev
aptitude install libxft-dev
```

2.3 root compilation - version 5.x

(example for 64bit)

I recommend to create three directories in $HOME - one containing root, the other for source files + compilation and the last for macros and compiled .so files:

```
mkdir ~/root
mkdir ~/root.inst
mkdir ~/root_macros
```

Append to your `$HOME/.bashrc` paths to your ~/root/:

```
export ROOTSYS=$HOME/root
export PATH=$ROOTSYS/bin:~/root_macros:$PATH
export LD_LIBRARY_PATH=$ROOTSYS/lib:$ROOTSYS/lib/root:$LD_LIBRARY_PATH
```

Download root: wget https://root.cern.ch/download/root_v5.34.34.source.tar.gz

Unpack to ~/root.inst and enter there

First you need to *configure* the Makefiles, if this is successful, use *make* and *make install*. To use more CPU cores and compare times you can do e.g. *time make -j4*

```
./configure linuxx8664gcc  --prefix=$HOME/root --etcdir=$HOME/root/etc --enable-opengl --en
```

```
make
```

```
make install
```

Root should reside in $HOME/root/bin and should be reachable from commandline: *root*.

Comment for Ubuntu xenial 16.04: *use root 5.34.36, edit ./root/tmva/src/RuleFitParams.cxx and change isnan(fstarVal) to std::isnan(fstarVal); root6 has still problem as of 2016/05*

3.1 shspe compilation

you may install the *~/.rootrc* and *~/.rootlogon* - it is my choice, you may prefer something differerent init_scripts/install_initscripts

```
cd  init_scripts
./install_initscripts
```

. . . this step can make some problems if you have already tuned these startup scripts, so check it before use.

and then go back to you shspe install directory (*cd ..*) and do

NEW

Now everything should be done with `cmake`

```
cmake .
cmake --build .
# make
# make install
```

shspe .so library should be installed in root macro directory (verify)

4.1 run root

```
root
```

and inside root shspe. It is a function so don't miss parentheses.

```
root [0] shspe()
```

4.1.1 AUTOLOAD

At this point - in the directory I want to work - I create *autoload_libs.C* with a content like this:

```
{
gROOT->ProcessLine("shspe()");
}
```

and the next start of root, the shspe() opens automaticaly.

4.1.2 .REMOTE_DATA_DIR

If the data are on different PATH than your working dir: create a /hidden/ file: `.REMOTE_DATA_DIR` . shspe() on *Openfile* will show the content also of this destination.

4.1.3 OPEN FILE

First thing you can do now is open a root file with histograms. Root files that are on `.REMOTE_DATA_DIR` path are displayed with a tilde ~. The actual opened file name written to /hidden/ `.CURRENTFILE`. This could be usefull if you needed to deduce the run number in some script or something. But this can also create difficulties when two shspe() sessions are open in the same directory.

4.1.4. OPEN FILE details

The shspe() can also open: * mysql reference files with a name like *table-name.mysql* /host-user-password/ that refers to database nemaed *test* * *mmap.histo file* that shares memory with histograms from digitizer acquisistion

4.1.5 FILE SAVE

You can save * canvases + histograms on screen click `SaveCanvas` - you can input you filename into the text field and save multiple format, with a time tag. Withou a name it stores canvas as 1.root, 2.root,... 9.root that can be recovered later with `LoadCanvas` * all histograms click `SaveAllSpectra` - filename is always based on datetime * copy histogram from file to memory `Spectrum2Memory` and go to memory later and compare histograms from different files there.

4.2 Fitting

4.2.1 Prepare fit

4.2.2 Fit procedure

4.2.3 Save fit

Several files are created `zfitresults` the extensions are

`.eff` - for calibration of efficiency `.ecal` - just energy and channel *`.tmp` - same as eff but only the last fit - can be parsed by script* `final.root` - all objects of results *`tmp.root` - as root, but the last fit

5.1 Other functions

There are more functions loaded with shspe:

gr__ group

functions to play with graphs

`grhelp()`

joingraphs

join several graphs to one multigraph

MPad group

MPadGetByName() MPadCreate() MPadPrintIn()

for use with TCounter

TCounter group

not interesting now

cuts group

uses **cuts.root** file for storing the cuts

```
cutload()
cutsave()
cutls()
cutrm()
cutcp()
```

6.1 some specific details

creates .CURRENTFILE on open

checks .REMOTE_DATA_DIR on open

searches for a script *shspe.pk_mysql* setup file on savefit

edited in http://dillinger.io/ and elpa-markup-more

7.0 root 6 compilation

- Good on 16.04 Ubuntu Xenial[1]
- tests on Debian 8 jessie:[2]

[1] Ubuntu 16.04 is LTS version until 2018/04
[2] 2017/06 - debian Stretch is out

- v6.06.06 - worked
- v6.09.02 - had to get backports `cmake` : `deb http://ftp.debian.org/debian jessie-backports main; aptitude -t jessie-backports upgrade cmake, aptitude install libglew-dev (i think)` compilation CRASHED, but root worked anyway
- tests on Debian 9 Stretch - `None`

To enable TF1H in pyroot (v6.09.02):

http://stackoverflow.com/questions/33361998/pyroot-attributeerror-while-importing-histograms/33363251

never make / cmake inside root-6.xx.xx directory!

7.1 commands to compile/install

1. `mkdir ~/root` and `mkdir root.build` :create extra root.build directory, `cd root.build`
2. configure: ending with `-DCMAKE_INSTALL_PREFIX=$HOME/root` and `-Dpython3="ON"` to set the installation dir and python3 for jupyter

```
cmake ../root-v6-10-02/    -DCMAKE_CXX_FLAGS="-D_GLIBCXX_USE_CXX11_ABI=0" -Dcxx14="ON" -Dall
```

3. `time cmake --build . -- -j4` :build - prepare for 1:30h with 4 cores laptop
4. `source bin/thisroot.sh` - it could/should/may be included in `.zshrc .bashrc`
5. `cmake --build . --target install` - install to ~/root/ directory

```
#export PYTHONPATH=$HOME/root/lib/
```

After, it is possible to return the line with PATH to `.bashrc` but most probably it puts thisroot.sh there.

7.2 problems

- with 4 cores, it did not compile (before 6.08.00 versions) - **OK NOW** for later
- when anaconda is installed, there was a mess Update: anaconda3 is UNTESTED
- last tests with Pro: 6-08-06, 6-09-02

- look at the page https://root.cern.ch/building-root#options

and then try (make was 132 minutes on one core):

- See https://root.cern.ch/building-root

  this must be in `cmake` *-DCMAKE-INSTALL_PATH=$HOME/root*

Statistics:

```
v6.09.02 laptop
j4=9730s user - 43m total;
8cores -9973s - 45:19total;
2cores - 6499s - 55:58 total

6.09.02 on aaron - 4 cores: crashed at 95%; but root worked
real    30m22.488s
user    112m15.488s
sys 3m55.304s
```

compile test:

- shspe should compile
- gregory mmap.histo should be opened by shspe (mmapfile)
- pyroot - TH1F should be imported into python