



githubAccess_public

Description:

This routine describes how to set up a github account and pull repositories of the CFDEM(R)project. After setting some environment variables LIGGGHTS(R) and CFDEM(R)coupling can be compiled

Procedure:

Basically the following steps have to be performed:

- *git clone* the desired repository
- update your repositories by *git pull*
- set environment variables
- compile LIGGGHTS(R) and CFDEM(R)coupling
- run your own cases

git clone the desired repository:

If not already done, open a terminal and create a directory for LIGGGHTS(R) in \$HOME:

```
cd  
  
mkdir LIGGGHTS  
  
cd LIGGGHTS
```

To clone the public LIGGGHTS repository, open a terminal and execute:

```
git clone git://github.com/CFDEMproject/LIGGGHTS-PUBLIC.git LIGGGHTS-PUBLIC
```

If not already done, open a terminal and create a directory for CFDEM(R)coupling in \$HOME:

```
cd  
  
mkdir CFDEM  
  
cd CFDEM
```

Make sure that OpenFOAM(R) is already set up correctly!

To clone the public CFDEM(R)coupling repository, open a terminal and execute:

```
git clone git://github.com/CFDEMproject/CFDEMcoupling-PUBLIC.git CFDEMcoupling-PUBLIC-$WM_PROJECT_VERSION
```

Troubles? See Troubleshooting section below.

Update your repositories by *git pull*:

To get the latest version, open a terminal, go to the location of your local installation and type: *Warning: git stash will remove your changes in \$HOME/CFDEM/CFDEMcoupling-PUBLIC-\$WM_PROJECT_VERSION !*

```
cd $HOME/CFDEM/CFDEMcoupling-PUBLIC-$WM_PROJECT_VERSION
git stash
git pull
```

Set Environment Variables:

Now you need to set some environment variables in ~/.bashrc (if you use c-shell, manipulate ~/.cshrc accordingly). Open ~/.bashrc

```
gedit ~/.bashrc &
```

add the lines (you find them also in .../cfdemParticle/etc/bashrc and cshrc respectively):

```
#=====#
#- source cfdem env vars
export CFDEM_VERSION=PUBLIC
export CFDEM_PROJECT_DIR=$HOME/CFDEM/CFDEMcoupling-$CFDEM_VERSION-$WM_PROJECT_VERSION
export CFDEM_SRC_DIR=$CFDEM_PROJECT_DIR/src
export CFDEM_SOLVER_DIR=$CFDEM_PROJECT_DIR/applications/solvers
export CFDEM_DOC_DIR=$CFDEM_PROJECT_DIR/doc
export CFDEM_UT_DIR=$CFDEM_PROJECT_DIR/applications/utilities
export CFDEM_TUT_DIR=$CFDEM_PROJECT_DIR/tutorials
export CFDEM_PROJECT_USER_DIR=$HOME/CFDEM/$LOGNAME-$CFDEM_VERSION-$WM_PROJECT_VERSION
export CFDEM_bashrc=$CFDEM_SRC_DIR/lagrangian/cfdemParticle/etc/bashrc
export CFDEM_LIGGGHTS_SRC_DIR=$HOME/LIGGGHTS/LIGGGHTS-PUBLIC/src
export CFDEM_LIGGGHTS_MAKEFILE_NAME=ubuntuVTK_fpic
export CFDEM_LPP_DIR=$HOME/LIGGGHTS/mylpp/src
export CFDEM_PIZZA_DIR=$HOME/LIGGGHTS/PIZZA/gran_pizza_17Aug10/src
. $CFDEM_bashrc
#=====#
```

Save the ~/.bashrc, open a new terminal and test the settings. The commands:

```
$CFDEM_PROJECT_DIR
$CFDEM_SRC_DIR
$CFDEM_LIGGGHTS_SRC_DIR
```

should give "...: is a directory" otherwise something went wrong and the environment variables in ~/.bashrc are not set correctly.

To specify the paths of pizza, please check the settings in \$CFDEM_SRC_DIR/lagrangian/cfdemParticle/etc/bashrc.

If \$CFDEM_SRC_DIR is set correctly, you can type

```
cfdemSysTest
```

to get some information if the paths are set correctly.

Compile LIGGGHTS(R) and CFDEM(R)coupling:

```
git clone git://github.com/CFDEMproject/CFDEMcoupling-PUBLIC.git CFDEMcoupling-PUBLIC-$WM_PROJ
```

If above settings were done correctly, you can compile LIGGGHTS(R) by typing:

```
cfdemCompLIG
```

and you can then compile CFDEM(R)coupling by typing:

```
cfdemCompCFDEM
```

You can run the tutorial cases by executing `.../etc/testTutorial.sh` through the alias `cfdemTestTUT`. Alternatively you can run each tutorial using the `Allrun.sh` scripts in the tutorial directories.

In case questions concerning the installation arise, please feel free to contact our forum at www.cfdem.com.

Run Your Own Cases:

If you want to run your own cases, please do so in `$CFDEM_PROJECT_USER_DIR/run` which is automatically being generated. E.g. copy one of the tutorial cases there, adapt it to your needs. Changes in `$CFDEM_TUT_DIR` will be lost after every *git stash*!

Additional Installations:

Optionally you can install lpp which will help you convert the DEM (dump) data to VTK format. For standard CFD-DEM runs this will not be necessary. To get the DEM postprocessing tool "lpp" you need python-numpy package installed:

```
sudo apt-get install python-numpy
```

You can pull the latest version of lpp with:

```
cd $HOME/LIGGGHTS
```

```
git clone git://cfdem.git.sourceforge.net/gitroot/cfdem/lpp mylpp
```

Backwards Compatibility:

Basically CFDEM(R)coupling supports one OpenFOAM(R) version therefore all settings are prepared for that. Nevertheless we try to maintain backwards compatibility as long as it works with reasonable effort.

The supported OpenFOAM(R) and LIGGGHTS(R) versions are stated in:
`src/lagrangian/cfdemParticle/cfdTools/versionInfo.H`

For using other versions you can manipulate: `src/lagrangian/cfdemParticle/etc/OFversion/OFversion.H` (still not all functionality might work then!)

Troubleshooting:

- Troubles with git clone?

a) The git protocol will not work if your computer is behind a firewall which blocks the relevant TCP port, you can use alternatively (write command in one line):

```
git clone git://github.com/CFDEMproject/CFDEMcoupling-PUBLIC.git CFDEMcoupling-PUBLIC-$WM_PROJECT
```

```
git clone https://user@github.com/CFDEMproject/CFDEMcoupling-PUBLIC.git  
CFDEMcoupling-PUBLIC-$WM_PROJECT_VERSION
```

b) If you face the error: "error: SSL certificate problem, verify that the CA cert is OK. Details: error:14090086:SSL routines:SSL3_GET_SERVER_CERTIFICATE:certificate verify failed while accessing https://github.com/...",

please use: `env GIT_SSL_NO_VERIFY=true git clone https://github...`

(see <http://stackoverflow.com/questions/3777075/https-github-access>)

c) If you face the error: "Agent admitted failure to sign using the key. Permission denied (publickey).", after `ssh -T git@github.com`

please type: "ssh-add"

(see: <https://help.github.com/articles/error-agent-admitted-failure-to-sign>)

```
git clone git://github.com/CFDEMproject/CFDEMcoupling-PUBLIC.git CFDEMcoupling-PUBLIC-$WM_PROJECT_VERSION
```