



CFDEM  
coupling

## githubAccess\_public

---

### Description:

This routine describes how to setup a github account and pull repositories of the CFDEMproject. After setting some environment variables LIGGGHTS and CFDEMcoupling 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 and CFDEMcoupling
- run your own cases

#### ***git clone* the desired repository:**

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

```
cd  
  
mkdir LIGGGHTS  
  
cd LIGGGHTS
```

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

```
git clone git://cfdem.git.sourceforge.net/gitroot/cfdem/liggghtsdev LIGGGHTS-PUBLIC
```

If not already done, open a terminal and create a directory for CFDEMcoupling in \$HOME:

```
cd  
  
mkdir CFDEM  
  
cd CFDEM
```

Make sure that OpenFOAM(R)-2.1.x is already set up correctly!

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

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

Note: the git protocol will not work if your computer is behind a firewall which blocks the relevant TCP port, you can use alternatively:

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

## 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:

```
#=====#
#- 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/lagrangian/cfdemParticle
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=$CFDEM_SRC_DIR/etc/bashrc
export CFDEM_LIGGGHTS_SRC_DIR=$HOME/LIGGGHTS/LIGGGHTS-PUBLIC/src
export CFDEM_LIGGGHTS_MAKEFILE_NAME=fedora_fpic
. $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/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 and CFDEMcoupling:

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

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

cfdemCompLIG

and you can then compile CFDEMcoupling 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](http://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*!