



githubAccess_public

Description:

This routine describes how to pull repositories of the CFDEM(R)project from [github.com](#). 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* and setup OpenFOAM
- *git clone* the desired repositories
- update your repositories by *git pull*
- set environment variables
- compile LIGGGHTS(R) and CFDEM(R)coupling
- run your own cases

git clone and setup OpenFOAM:

Have a look at the latest compatible OpenFOAM(R)-version in the versionInfo.H file at [github](#). This file will later be downloaded as a part of the source-code. Look for the git commit hashtag in the following line:

```
word OFversion="<OF-Release>-commit-<commitHashtag>"
```

```
e.g. word OFversion="2.4.x-commit-3d8da0e960c717ff582f1517a27724144f086b83"
```

However sometimes even newer versions are supported, please check the [release notes](#) and the "Advanced Settings"-section.

Basically follow the OpenFOAM(R) git compilation [instructions](#), with a small number of exceptions:

When you git clone the repository, replace the release-version with <OF-Release>.

with git-protocol:

```
git clone git://github.com/OpenFOAM/OpenFOAM-<OF-Release>.git
```

or with https:

```
git clone https://github.com/OpenFOAM/OpenFOAM-<OF-Release>.git
```

Now change into the new directory and checkout the correct compatible version:

```
cd OpenFOAM-<OF-Release>
git checkout <commitHashtag>
```

The result will be a status report, that indicates a 'detached head state'. Now continue with installing and compiling OpenFOAM(R). Make sure that OpenFOAM(R) works properly with a parallel Simulation!

If you want to use an older OpenFOAM(R)-version, please have a look at the "Backwards Compatibility"-section.

git clone the desired repositories:

You may want to take a look around on CFDEMproject on github: github.com/CFDEMproject_gitCFDEM

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: with git-protocol:

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

or with https:

```
git clone https://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
```

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

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

or with https:

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

Additionally the lpp tool for converting LIGGGHTS dump-files into the paraview readable VTK-format with git-protocol:

```
git clone git://github.com:CFDEMproject/LPP.git $HOME/LIGGGHTS/mylpp
```

or with https:

```
git clone https://github.com:CFDEMproject/LPP.git $HOME/LIGGGHTS/mylpp
```

Please have a look at README and INSTALL.txt in the root directory of LPP for further information.

Troubles? See Troubleshooting git 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 CFDEMCoupling-\$WM_PROJECT_VERSION/src/lagrangian/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=fedora_fpic
export CFDEM_LPP_DIR=$HOME/LIGGGHTS/mylpp/src
export CFDEM_PIZZA_DIR=$HOME/LIGGGHTS/PIZZA/gran_pizza_17Aug10/src
. $CFDEM_bashrc
#=====#
```

If you installed LIGGGHTS(R) or CFDEM(R)coupling in non-standard paths, please have a look at least at CFDEM_PROJECT_DIR and CFDEM_LIGGGHTS_SRC_DIR. The standard CFDEM_LIGGGHTS_MAKEFILE_NAME is fedora_fpic, which works on most systems. However please checkout LIGGGHTS-PUBLIC/src/MAKE for additional makefiles, which are available. The most used ones are fedora_fpic and ubuntuVTK_fpic. Beware that the CFDEMCoupling needs a fpic compilation to use LIGGGHTS as a library. Please check the "Advanced Settings" for VTK information.

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:

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
```

or compile both at once with:

```
cfdemCompCFDEMA11
```

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
```

with git-protocol: `git clone git://github.com/CFDEMproject/LPP.git` with https: `git clone https://github.com/CFDEMproject/LPP.git`

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!)

Advanced Settings:

```
githubAccess_public
```

Here some advanced settings and hints for non-standard compilations are presented. As stated in the "Backwards Compatibility"-section, there are compiler flags for different OpenFOAM versions. Checkout `src/lagrangian/cfdemParticle/etc/OFversion/OFversion.H` for compatibility settings. Just comment the current `"#define version2X"` in the top-section and uncomment the one you want to compile it with.

There are advanced compilation settings for library-paths, includes and libraries are within the `additionalLibs` file in `src/lagrangian/cfdemParticle/etc/additionalLibs`. There are predefined files for different OpenFOAM versions. To use a different version, add the following lines to your `.bashrc` (`.cshrc`) before the standard CFDEM variables:

```
export CFDEM_ADD_LIBS_DIR=FOLDER_OF_NEW_additionalLibs_FILE/

export CFDEM_ADD_LIBS_NAME=additionalLibs30x
```

This is an example to use a predefined `additionalLibs` file for OpenFOAM-3.0.x.

To enable direct VTK-dump (dump custom/vtk) support of LIGGGHTS and CFDEMcoupling, you have to install the VTK libraries. Either 5.8 or 6 are predefined for ubuntu.

```
sudo apt-get libvtk5.8 libvtk5-dev
```

Change `"export CFDEM_LIGGGHTS_MAKEFILE_NAME=fedora_fpic"` in your `.bashrc` according to your preferred LIGGGHTS makefile. If you have a non-standard installation location you have to adapt the LIGGGHTS makefile accordingly. To enable this feature in a coupled run the `additionalLibs` file has to be modified. It basically needs to include the same libraries as the LIGGGHTS-Makefile. E.g. for Ubuntu-14.04 with vtk-5.8:

```
CFDEM_ADD_LIB_PATHS = -L/usr/include/vtk-5.8

CFDEM_ADD_LIBS      = -lvtkCommon -lvtkFiltering -lvtkIO
```

Troubleshooting git:

- 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 https instead of git (write command in one line):

```
git clone https://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>)