Project
openfoam
Manage
Plan
Issues
Issue boards
Milestones
Wiki
Code
Deploy
Analyze

configuring

Last edited by **Mark Olesen** 11 months ago

The main OpenFOAM settings are located in the parent `etc/` directory with both POSIX (bash, dash,...) and csh shells being supported. To use OpenFOAM, source either the `etc/bashrc` or the `etc/cshrc` file, as appropriate.

Configuration layers

Before launching into manually adjusting the configuration, it is useful to first understand how OpenFOAM supports different configuration *layers*. Similar to file-system permissions, we use the notion of **user**, **group**, **other** categories when searching for files. The output of `foamEtcFile` can be used to obtain a quick overview:

```
$ foamEtcFile -list

$HOME/.OpenFOAM/2406
$HOME/.OpenFOAM
/path/OpenFOAM-v2406/site/2406/etc
/path/OpenFOAM-v2406/site/etc
/path/OpenFOAM-v2406/etc
```

Both the *user* paths (located as `$HOME/.OpenFOAM/`) and the *group* paths (`/path/OpenFOAM-v2406/site/`) support additional API versioning to allow different settings between releases. The **other** corresponds to the settings shipped with a particular OpenFOAM release.

Making configuration changes under the *user* or *group* directories allows you to preserve these across upgrades and makes it easier (if necessary) to revert to the original values!!

Making changes to the configuration

The first encounter with the OpenFOAM configuration files can be somewhat intimidating. There are indeed quite a few different bits of software related to using OpenFOAM, each of which could be available in different preferred versions, in different possible locations and with different conventions for naming their library directories. Additionally it should allow individual users to make their own configuration choices. Supporting cshell variants for everything adds yet more files to the mix. Fortunately, the user often only needs to make a few simple changes and can ignore most of the details and we also provide a `bin/tools/foamConfigurePaths` tool to make multiple common changes directly from the command line. The configuration files generally contain detailed information about which values they expect, and the user editable part is also clearly marked as such. For example,

```
#------------------------------------------------------------------------------
# USER EDITABLE PART: Changes made here may be lost with the next upgrade

ParaView_VERSION=5.10.1
ParaView_QT=qt-system

# END OF (NORMAL) USER EDITABLE PART
#------------------------------------------------------------------------------
```

Nonetheless, before making changes it can be useful to understand where these changes should actually be made (and why). To simplify things, we only discuss POSIX (bash), but most points apply to cshell variants as well.

1. The main entry point for the OpenFOAM configuration is the `etc/bashrc` file. The initial portion of the file establishes the version and contains some script *magic* to help us determine where the OpenFOAM directory is located. The balance of the file contains some general OpenFOAM-specific settings, which you can use for guidance but in general you should note the following:
   - Changes made to this `etc/bashrc` file ***will be lost*** with the next upgrade.
   - Overrides should defined in the `etc/prefs.sh` instead. See the comments section of the `etc/bashrc` file for more details.
2. The `etc/bashrc` file (our entry point) passes control to the `etc/config.sh/setup` file, which dispatches the rest of the configuration actions.

The setup of the OpenFOAM environment can be described in terms of a processing tree:

```
source etc/bashrc  [args]
       |
       |-- constants
       |-- directory discovery magic
       |-- defaults
       |-- define OpenFOAM directory
       |
       \-- setup
           |-- discovery of ThirdParty locations
           |-- admin overrides (prefs.sh file)
           |-- user overrides (prefs.sh file)
           |-- user overrides (arguments)
           |-- settings (compiler, os)
           |-- mpi
           |-- paraview
           |-- vtk / mesa (llvm)
           |-- CGAL / boost
           |-- scotch
           |-- FFTW
           \-- aliases
```

At most locations in this process it is possible for the user to influence the values used by providing an alternative version of the file. For example, simply creating the file `$HOME/.OpenFOAM/config.sh/FFTW` will cause it to be found by the `foamEtcFile` mechanism during sourcing (see `foamEtcFile -list` for a reminder of which directories will be searched). Most fairly permanent changes that affect the base configuration of OpenFOAM itself (choice of compiler, mpi, data sizes, etc) should normally be defined in the `prefs.sh` file. These type of changes are important enough that they receive special treatment. Use the base or admin `prefs.sh` file if available as `PROJECT/etc/prefs.sh`. This provides the system admin a reliable location to define site-wide settings, such as for compiler and vendor-specific MPI libraries. use the user or group prefs.sh if it exists. For quick or temporary changes, the special interpretation of arguments when sourcing the etc/bashrc are quite convenient. This mechanism allows direct setting of variables without needing to edit any files. For example, to source the OpenFOAM environment with a different compiler:

```
source /path/to/OpenFOAM-v2406  WM_COMPILER=Clang
```

If the argument does not appear to be an assignment of a variable, it will attempt to resolve it as a file and then source that. This property lets the user bundle some favourite settings and temporarily switch to them. For example, by creating a few predefined configurations:

```
# file = $HOME/.OpenFOAM/gcc82
export WM_COMPILER_TYPE=ThirdParty
export WM_COMPILER=Gcc82
export WM_LABEL_SIZE=32
```

or

```
# file = $HOME/.OpenFOAM/clang50-int64
export WM_COMPILER_TYPE=ThirdParty
export WM_COMPILER=Clang50
export WM_LABEL_SIZE=64
```

It is then possible to easily switch between different configurations:

```
source /path/to/OpenFOAM-v2406  clang50-int64
source /path/to/OpenFOAM-v2406  gcc82
source /path/to/OpenFOAM-v2406  wingw
```

Armed with this information, the user should be able to make adjustments to the OpenFOAM configuration with a good degree of confidence. However, there are also times in which it can be expedient and useful to simply change the entries directly within the OpenFOAM directory as new permanent defaults for all users. This can also be the case for cluster installations where the user will not require the usual flexibility. For these cases, the `bin/tools/foamConfigurePaths` tool can be helpful (and powerful). For example, when installing without any OpenFOAM ThirdParty dependencies and additionally setting the OpenFOAM directory to a fixed location (removing any bash discovery magic):

```
bin/tools/foamConfigurePaths \
    -project-path "/opt/openfoam2406" \
    -boost boost-system \
    -cgal  cgal-system \
    -fftw  fftw-system \
    -kahip kahip-none \
    -scotch scotch-system \
    -scotch-path /usr/lib64/mpi/gcc/openmpi \
    ;
```

Using this tool has some restrictions:

- It must be called from the OpenFOAM project directory
- It is not available in the PATH, since it we wish to avoid any inadvertent use
- Using this tool to change default gcc, gmp, mpfr versions is not very precise. It will change the gcc version without distinguishing between Gcc48, Gcc82 etc.

Pages 80
Search pages
building
cross compile mingw
coding
git workflow
patterns
dictionary
HashTable
memory
parallel
patterns
precision
registry
selectors
strings
scripts
scripts
style
filenames
style
configuring
Home
icons
info
modules
visualization
packaging
debian
Locations
README
sourceforge
README
suse
Locations
page access code
page build code
page feature requests
precompiled
apptainer
debian
docker
docker old
redhat
suse
windows
running
openfoam selector
shell session
Submitting issues
tuning
upgrade
upgrade
v3 Developer Upgrade G...
v3 User Upgrade Guide
v1606 Developer Upgrad...
v1606 User Upgrade Gui...
v1612 Developer Upgrad...
v1612 User Upgrade Guide
v1612 utility postProcess
v1706 Developer Upgrad...
v1706 User Upgrade Guide
v1712 Developer Upgrad...
v1712 User Upgrade Guide
v1806 Developer Upgrad...
v1806 User Upgrade Gui...
v1812 Developer Upgrad...
v1812 User Upgrade Guide
v1906 Developer Upgrad...
v1906 User Upgrade Gui...
v1912 Developer Upgrad...
v1912 User Upgrade Guide
v2006 Developer Upgra...
v2006 User Upgrade Gui...
v2012 Developer Upgrad...
v2012 User Upgrade Gui...

Working in groups

When an OpenFOAM cluster installation is being used by several different people or interest groups it can be highly interesting to share common setups or custom libraries and applications. This is where the OpenFOAM site (group) configuration can be quite helpful. The directory location of OpenFOAM site settings is defined by the `$WM_PROJECT_SITE` environment variable. If this is undefined, the default is to use `PROJECT/site` (ie, a site directory located within the OpenFOAM directory). Within this `$WM_PROJECT_SITE` directory, we can use a directory structure that mirrors elements of the OpenFOAM directory structure, but which also includes a degree of versioning as well:

```
$WM_PROJECT_SITE
|
|-- API
|   |-- bin
|   \-- etc
|-- VERSION
|   \-- platforms
|       |-- bin
|       \-- lib
|-- bin
\-- etc
```

Useful OpenFOAM-related scripts can be placed in the bin directory. If the script can only work with a particular OpenFOAM version, it then makes sense to place it into the API/bin directory accordingly. Similarly, if particular configurations or setups are useful for several people, it makes sense to locate them centrally as a site (or group) resource. For example,

```
$WM_PROJECT_SITE
|
\-- etc
    |-- caseDicts
    \-- config.sh
        |-- openmpi
        \-- paraview
```

for some jointly useful caseDicts and suitable configurations for openmpi, paraview. The `foamEtcFile -list` option provides a good overview of which locations will be searched for configuration files, which uses the following precedence:

- user:
  - `$HOME/.OpenFOAM/API`
  - `$HOME/.OpenFOAM`
- group:
  - `$WM_PROJECT_SITE/API/etc`
  - `$WM_PROJECT_SITE`
- other:
  - `$WM_PROJECT_DIR/etc`

If applications and libraries are to be shared within a group, a typical approach is that one person is in charge of administering the the internal code releases. They would compile the code in their normal user directories, which means that it would normally have the user destinations:

```
$FOAM_USER_APPBIN
$FOAM_USER_LIBBIN
```

For distribution at the group level, these files would be synchronized to the corresponding group directories:

```
$FOAM_USER_APPBIN  ->  $FOAM_SITE_APPBIN
$FOAM_USER_LIBBIN  ->  $FOAM_SITE_LIBBIN
```

Additional information

The `bashrc` or `cshrc` files source the following files in the `config.sh/` or `config.csh/` directories:

- `setup` : finalize setup of OpenFOAM environment (called by bashrc,cshrc)
- `settings` : core settings
- `aliases` : aliases for interactive shells
- `unset` : sourced to clear as many OpenFOAM environment settings as possible
- `mpi` : MPI communications library settings
- `paraview` : application settings for ParaView
- `scotch` : application settings for compiling against scotch
- `metis` : application settings for compiling against metis

The `config.{csh,sh}/example/` directories contain additional example configuration files for the corresponding shell:

- `compiler` : an example of fine tuning ThirdParty compiler settings
- `openmpi` : an example of fine tuning openmpi settings for OpenFOAM
- `paraview` : an example of chaining to the standard config/paraview with a different ParaView_VERSION
- `prefs.csh`, `prefs.sh`: examples of supplying alternative site-defined settings

Comments

Please register or sign in to add a comment.

v2106 Developer Upgrad...
v2106 User Upgrade Gui...
v2112 Developer Upgrad...
v2112 User Upgrade Guide
v2206 Developer Upgra...
v2206 User Upgrade Gui...
v2212 Developer Upgrad...
v2212 User Upgrade Guide
v2306 Developer Upgra...
v2306 User Upgrade Gui...
v2312 Developer Upgrad...
v2312 User Upgrade Gui...
v2406 Developer Upgra...
v2406 User Upgrade Gui...
v2412 Developer Upgrad...
v2412 User Upgrade Guide
View all pages

Help