

## Sample Jobs

Last update: Jan 30, 2025.

## Introduction

Sample input and jobscript are also available for each installed application. Those samples can also be used as a template file for your jobs. In this page, we will show you procedures how to run those sample jobs using GAMESS and Gromacs examples. As for Gaussian, please check [the special way using g16sub/g09sub](#) first.

## List of Installed Applications

There are several ways to know which applications are installed.

### 1. at web page

You can see the application list [in this page](#).

### 2. module avail

"module avail" command will give you a (long) list of applications. Applications which should be submitted as jobs would be listed in **"/apl/modules/apl" category, which is colored with red in the box below**

```
[user@ccfep3 ~]$ module avail
----- /apl/modules/defaults -----
2022

----- /apl/modules/oneapi -----
compiler-rt/2022.0.2 intelmpi/2021.7.1 mkl/2022.0.2 tbb/2021.7.1
compiler-rt/2022.2.1 intelpython/2022.0.0 mkl/2022.2.1
...
(中略)
...

----- /apl/modules/apl -----
amber/20u13 gromacs/2021.4-CUDA nwchem/6.8
amber/22u1 gromacs/2021.6 nwchem/7.0.2
cp2k/9.1 gromacs/2021.6-CUDA openmolcas/v21.10
cp2k/9.1-impi gromacs/2022.4 openmolcas/v22.10
crystal/17-1.0.2 gromacs/2022.4-CUDA orca/4.2.1
gamess/2021R1 GRRM/14-g09 orca/5.0.3
gamess/2022R2 GRRM/17-g09 qe/6.8
gaussian/09e01 GRRM/17-g16 qe/6.8-gpu
gaussian/16b01 lammmps/2021-Sep29 reactionplus/1.0
gaussian/16c01 lammmps/2021-Sep29-CUDA siesta/4.1.5-mpi
gaussian/16c02 lammmps/2022-Jun23 siesta/4.1.5-omp
genesis/2.0.3 lammmps/2022-Jun23-CUDA turbomole/7.6-mpi
genesis/2.0.3-CUDA namd/2.14 turbomole/7.6-serial
gromacs/2021.4 namd/2.14-CUDA turbomole/7.6-smp

Key:
loaded directory/ auto-loaded default-version modulepath
[user@ccfep3 ~]$
```

(Press "q" key or scroll to the bottom to quit "module avail" command.)

### 3. see /apl directly

Non-OS standard applications and libraries are installed under /apl. You can directly check the list there.

```
[user@ccfep3 /apl]$ ls
amber crystal gaussian hpc-x mvapich nwchem orca reactionplus
```

```

aocc cuda genesis lammmps namd oneapi pbs siesta
aocl dirac gromacs modules nbo openmolcas psi4 turbomole
cp2k gamess GRRM molpro nvhpc openmpi qe vmd
[user@ccfep3 /apl]$ ls /apl/amber
20u13 22u1
[user@ccfep3 /apl]$ ls /apl/amber/20u13
amber.csh build configure lib64 README test
amber.sh cmake dat logs recipe_at update_amber
AmberTools CMakeLists.txt doc Makefile samples updateutils
benchmarks cmake-packaging include miniconda share wget-log
bin config.h lib miniforge src
[user@ccfep3 /apl]$

```

## Location of Sample Files

Sample input for a certain application is generally available under `/apl/(application name)/(version/revision)/samples` directory.

Example: check sample directory of gromacs 2021.6.

```

[user@ccfep3 /apl]$ ls /apl
amber crystal gaussian hpc-x mvapich nwchem orca reactionplus
aocc cuda genesis lammmps namd oneapi pbs siesta
aocl dirac gromacs modules nbo openmolcas psi4 turbomole
cp2k gamess GRRM molpro nvhpc openmpi qe vmd
[user@ccfep3 /apl]$ ls /apl/gromacs/
2021.4 2021.4-CUDA 2021.6 2021.6-CUDA 2022.4 2022.4-CUDA
[user@ccfep3 /apl]$ ls /apl/gromacs/2021.6
bin include lib64 samples share
[user@ccfep3 /apl]$ ls /apl/gromacs/2021.6/samples/
conf.gro sample-mpi.csh sample-threadmpi.csh topol.top
grompp.mdp sample-mpi.sh sample-threadmpi.sh

```

(package name with "-CUDA" is GPU-enabled version.)

## Files in Sample Directory

In a sample directory, there is only one input data set in principle. However, there can be several job scripts in a sample directory (same input but using different shell, hardware, setting method).

Examples:

- `sample.sh => /bin/sh sample script`
- `sample.csh => /bin/csh sample script`
- `sample-gpu.sh => /bin/sh sample script using GPU`

Reading and comparing those files might be helpful to you.

### Example: gamess 2022R2

There are three scripts (`sample.csh`, `sample-module.sh`, `sample.sh`) for GAMESS 2022R2.

```

[user@ccfep4 ~]$ ls /apl/games/2022R2/samples/
exam01.inp sample.csh sample-module.sh sample.sh

```

### Example: gromacs 2021.6

There are four different scripts for Gromacs 2021.6.

```

[user@ccfep4 samples]$ ls
conf.gro sample-mpi.csh sample-threadmpi.csh topol.top
grompp.mdp sample-mpi.sh sample-threadmpi.sh

```

- `-mpi => parallel version with Open MPI (HPC-X); multinode parallel possible.`
- `-threadmpi => thread MPI parallel version; multinode parallel not available.`

## Run Sample: Basics

- copy files to your directory.
- "cd" to the directory, where the copied files exist
- submit a job (e.g. jsub sample.sh)
- Optional: usually, you can run samples on the login servers (e.g. sh ./sample.sh)
  - The way of # of CPUs specification might be different between "jsub" and "sh" cases.
  - GPU runs are not possible on ccfe. Please login to ccgpu from ccfe ("ssh ccgpu" command).
  - ccgpu is equipped with two GPU cards. MPI-parallel tests are also possible.

### Example 1: gamess2022R2

We here assume your test directory is ~/gamess2022R2\_test.

```
[user@ccfe4 ~]$ mkdir -p ~/gamess2022R2_test
[user@ccfe4 ~]$ cd ~/gamess2022R2_test
[user@ccfe4 gamess2022R2_test]$ cp /apl/gamess2022R2/samples/* .
[user@ccfe4 gamess2022R2_test]$ ls
exam01.inp sample-module.sh sample.csh sample.sh
[user@ccfe4 gamess2022R2_test]$ jsub sample.sh
4008689.cccms1
```

Status of submitted job can be checked with "jobinfo -c".

```
[user@ccfe4 gamess2022R2_test]$ jobinfo -c
-----
Queue Job ID Name      Status CPUs User/Grp  Elaps Node/(Reason)
-----
H      4008689 sample.sh  Run     4 user/--  -- ccc001
-----
[user@ccfe4 gamess2022R2_test]$
```

If the system is not terribly crowded, the job will soon finish and you can get the result.

```
[user@ccfe4 gamess2022R2_test]$ ls ~/gamess2022R2_test
exam01.dat exam01.log sample-module.sh sample.sh.e4008689
exam01.inp sample.csh sample.sh      sample.sh.o4008689
[user@ccfe4 gamess2022R2_test]$
```

Reference: sample.csh ([Explanations colored with blue do not exist in the original file](#))

```
#!/bin/sh
#PBS -l select=1:ncpus=4:mpiprocs=4:ompthreads=1 # <= 4-core job (in a vnode)
#PBS -l walltime=24:00:00 # <= time limit of this jobs is 24 hours

if [ ! -z "${PBS_O_WORKDIR}" ]; then
  cd ${PBS_O_WORKDIR} # <= cd to directory where you submit job (standard action for PBS jobs)
  NCPUS=$(wc -l < ${PBS_NODEFILE})
else
  NCPUS=4 # <= these two lines are setting for non-queuing system run
  export OMP_NUM_THREADS=1
fi

module -s purge
module -s load intelmpi/2021.7.1 # <= load required packages; depend on application
module -s load compiler-rt/2022.2.1

# processes per node; equal to mpiprocs value
PPN=4 # <= PPN = process per node; set to the mpiprocs value defined in the beginning

VERSION=2022R2
RUNGMS=/apl/gamess/${VERSION}/rungms
INPUT=exam01.inp
```

```
#{RUNGMS} ${INPUT:r} 00 $NCPUS $PPN >& ${INPUT%.*}.log # <= run GAMESS here
```

## Example 2: gromacs 2021.6

We here assume your test directory is ~/gromacs2021.6\_test.

```
[user@ccfep4 ~]$ mkdir -p ~/gromacs2021.6_test
[user@ccfep4 ~]$ cd ~/gromacs2021.6_test
[user@ccfep4 gromacs2021.6_test]$ cp /apl/gromacs/2021.6/samples/* .
[user@ccfep4 gromacs2021.6_test]$ ls
conf.gro  sample-mpi.csh  sample-threadmpi.csh  topol.top
grompp.mdp  sample-mpi.sh  sample-threadmpi.sh
[user@ccfep4 gromacs2021.6_test]$ jsub sample-mpi.sh
4008695.ccpbs1
```

Status of submitted job can be checked with "jobinfo -c".

```
[user@ccfep3 gromacs2021.6_test]$ jobinfo -c
-----
Queue Job ID Name      Status CPUs User/Grp  Elaps Node/(Reason)
-----
H      4008695 sample-mpi.sh Run      6 user/--- -- ccc001
-----
```

If the system is not terribly crowded, the job will soon finish and you can get the result.

```
[user@ccfep4 gromacs2021.6_test]$ ls ~/gromacs2021.6_test
conf.gro  md.log      sample-mpi.sh.e4008695  topol.top
confout.gro  mdout.mdp  sample-mpi.sh.o4008695  topol.tpr
ener.edr  mdrun.out  sample-threadmpi.csh
grompp.mdp  sample-mpi.csh  sample-threadmpi.sh
grompp.out  sample-mpi.sh  state.cpt
[user@ccfep4 gromacs2021.6_test]$
```

Reference: sample-mpi.sh ([Explanation colored with blue do not exist in the original file](#))

```
#!/bin/sh
#PBS -l select=1:ncpus=6:mpiprocs=6:ompthreads=1 # <= 6 core jobs (6 MPI processes)
#PBS -l walltime=00:30:00 # <= time limit is 30 minutes

if [ ! -z "${PBS_O_WORKDIR}" ]; then
  cd "${PBS_O_WORKDIR}"
fi

# non-module version
./apl/hpc-x/2.13.1/hpcx-rebuild-gcc11.sh # <= load Open MPI (HPC-X) environment
hpcx_load
export LD_LIBRARY_PATH="/apl/pbs/22.05.11/lib:${LD_LIBRARY_PATH}"
./apl/gromacs/2021.6/bin/GMXRC # <= load Gromacs related setting

### module version
#module -s purge
#module -s load --auto gromacs/2021.6 # <= setting above is also defined in this module

#####

N_MPI=6
N_OMP=1

gmx grompp -f grompp.mdp >& grompp.out
mpirun -v -n ${N_MPI} gmx_mpi mdrun -ntomp ${N_OMP} -s topol >& mdrun.out
```

## Tips about job scripts

You can find some examples of job header lines in [this page](#).

