

# **GAMESS**

## **Table of Contents**

- [Introduction](#)
- [Licensing and access](#)
- [Running GAMESS on Grex](#)
- [Running GAMESS on Jasper](#)
- [Running GAMESS on Lattice and Parallel](#)

## **Introduction**

The General Atomic and Molecular Electronic Structure System (GAMESS) is a molecular *ab initio* and DFT quantum chemistry package. It has a wide variety of high-level correlated wavefunction methods, recent Density Functionals, and several approximate methods for studying large systems, such as Fragment Molecular Orbital, Effective Fragment Potentials and others. The strengths of the code are in parallel computations using Distributed Memory approach. See the [GAMESS home page](#) [1] for documentation. The version installed on WestGrid systems is the one maintained by the Gordon group at Iowa State University, not [GAMESS-UK](#) [2].

GAMESS has been used by a number of WestGrid researchers, some installing it in their own directories. WestGrid administrators have installed GAMESS for general use on Bugaboo, Grex, Hungabee, Jasper, Lattice and Parallel. See system-specific notes for some of those systems below. See the Software Versions tab for a summary of the versions installed on each system.

On several systems the GAMESS environment is configured with the module command (as illustrated in the sample scripts below). Type

```
module avail
```

to see a list of available versions, then, change the module load ... line your script to load the specific version you want. See the WestGrid [modules page](#) [3] for more information about modules.

## **Licensing and access**

We are allowed to expose the executables, documentation and test files to our users, but, access to the source code is restricted.

Please read the [license conditions](#) [4] on the GAMESS website before using the software on WestGrid.

## Running GAMESS on Grex

On Grex (grex.westgrid.ca), the GAMESS environment is configured with the **module** command. The most recent (Aug 2016 R1, Dec. 2014 R1 and May 2013) versions use OpenMPI. This is the recommended version for most of the users, as it scales better and is more reliable; only for backward compatibility reasons, an old "sockets" version (May 2012 R1 ) is retained.

The GAMESS environment is accessible via the module load gamess/VERSION, and the command to start GAMESS computation is `rungms` script. For batch jobs, both `-l procs=xxx` and `-l nodes=yyy:ppn=zzz` styles are possible for specifying the CPU resources in the TORQUE batch job scripts; there is no requirement for requesting of the whole Grex nodes. The more flexible `-l procs` syntax is recommended. Input files for GAMESS usually have the `.inp` extension (as in `exam01.inp`).

Behavior of the **rungms** script can be influenced by certain environment variables. By default, the script attempts to use local scratch directories on the nodes, which is the best choice in most cases. If there is not enough space there, one can set `GLOBAL_SCRATCH` environment variable. It will place temporary GAMESS files under `/global/scratch/$USER/gamess-scratch`. In order to specify custom basis sets, the `EXTBAS` environment variable can be set.

An example of job script for parallel, Open MPI version of GAMESS on Grex is provided below. Note that it uses 32 CPU cores, a large amount (4Gb) of memory per process, and the local scratch (`GLOBAL_SCRATCH` is not present -- this is recommended). A custom basis is supplied with the `EXTBAS` directive; you can comment that out if using only built-in basis sets.

```
#!/bin/bash
#PBS -l procs=32,pmem=4gb,walltime=20:00:00
#PBS -m bea
#PBS -N CCSDT_dpi_rest
#PBS -r n

module purge
module load gamess/Dec2014

cd $PBS_O_WORKDIR
echo "starting GAMESS at " `date`

export EXTBAS=/global/scratch/gshamov/devel/gamess-test/tests/laikov.basis
rungms large_ccsd.inp 02 > large_ccsd.log

echo "all done! at" `date`
```

Note that in the example above, the version number parameter to `rungms` is set to 02 (which

is the default) and number of CPUs (the third parameter) is omitted. Open MPI jobs will get the number of CPUs from TORQUE automatically. If running interactive jobs, the number of CPUs should be provided explicitly:

```
module load gamess/Dec2014
rungms exam01 02 | tee exam01.log
```

Note that interactive use is only for small test jobs (for example those with RUNTYP=CHECK) and not for production computations, which should be done only using batch mode as described above.

The rungms script will place GAMESS formatted restart files (PUNCH, TRAJCT, etc. -- for a given *input\_name.inp* they will be *input\_name.dat*, *input\_name.trj*, ..) in the *./scr* subdirectory. The directory will be created automatically. Note that GAMESS will abort the current job if it finds the restart files already present there.

## Running GAMESS on Jasper

On Jasper ([jasper.westgrid.ca](http://jasper.westgrid.ca)), the GAMESS environment is configured with the module command.

It uses MPI for internode communication rather than sockets and either 'procs' or 'nodes:ppn' form can be used for requesting cores. It is recommended to use the 'procs' form as it would tend to run sooner.

A minimal script for a parallel job using the 01May2012 GAMESS code is:

```
#!/bin/bash
#PBS -S /bin/bash
#PBS -l procs=xxxx
#PBS -l pmem=2gb
#PBS -l walltime=24:00:00
module load application/gamess/20120501
cd $PBS_O_WORKDIR
ncpus=$(wc -l $PBS_NODEFILE | awk '{print $1}')
rungms inputfile 01 $ncpus
```

## Running GAMESS on Lattice and Parallel

**NOTE:** GAMESS is not currently available on Lattice and Parallel, so, choose one of the other WestGrid systems on which to run it. The notes below relate to a version that has been removed from Lattice and Parallel. Sorry for the inconvenience.

Lattice and Parallel share software directories, so, although the instructions below were originally written for Lattice, they should also apply for Parallel. Also, note that accounts on Lattice and Parallel are not automatically set up when you are first approved for a WestGrid account. See the [Lattice](#) <sup>[5]</sup> and [Parallel](#) <sup>[6]</sup> QuickStart Guides for instructions on applying for accounts on these systems.

GAMESS has been configured on Lattice for MPI parallel execution. Good parallel

performance for appropriate types of calculations and problem sizes should be expected. For example, a common Taxol benchmark scales well to at least 32 processors.

GAMESS on Lattice makes use of the MVAPICH 1.2 MPI-1 environment. To properly allocate processes, TORQUE jobs must be run using the `rungms` script provided. This script takes into account the GAMESS behavior of running one data communication and one computational process on each processor. The data communication process is lightweight and contributes little to the processor load. As noted in the [Lattice QuickStart Guide](#) [7], it is recommended that jobs on Lattice request whole nodes by using a TORQUE resource request of the form `-l nodes=nn:ppn=8`, with the processors per node parameter (**ppn**) equal to 8 and `nn` equal to the number of nodes. Do not use the **procs** resource request or the `rungms` script that is used to start GAMESS may not function correctly.

GAMESS scratch files are created in a sub-directory of the user's `/global/scratch/$USER/gamess-scr` directory. Each job uses a unique scratch directory named with the TORQUE job number. The scratch files are removed following each job whether it succeeds or fails, however it is good practice to check for scratch files following a run which terminates abnormally.

The `rungms` script writes important output files to the directory from which the job was submitted. The script will exit immediately, with a warning, if files exist from a previous run. As an example, copy the Taxol GAMESS input files to a `gtest` directory:

```
mkdir ~/gtest
cd ~/gtest
cp /global/software/gamess/gamess2010oct01/sample/* .
```

The file `taxol.pbs` contains:

```
#!/bin/bash
GMS_PATH="/global/software/gamess/gamess2010oct01"
VER="03"
JOBNUM=$(basename $PBS_JOBID .hn001)
cd ${PBS_O_WORKDIR}
${GMS_PATH}/rungms taxol-6-31g ${VER} &> taxol-6-31g.${JOBNUM}.log
```

Note the version number 03 corresponds to the R3 release of GAMESS 2010-Oct-01, which is also the default. To submit the job using 24 processors (3 nodes with 8 processors per node):

```
qsub -l walltime=1:00:00 -l nodes=3:ppn=8 -l mem=30gb taxol.pbs
```

Adjust the `mem` parameter according to the number of whole nodes requested, requesting 10gb per node. A copy of the output from this sample run is in `taxol-6-31g.sample.log`.

---

Updated:

2013-12-06 - Removed Checkers instructions.

2015-12-09 - Removed out-dated GAMESS from Lattice and Parallel.

### For More Information

<http://www.msg.ameslab.gov/index.html>

<http://www.msg.ameslab.gov/index.html> [1]

**Category**

[Electronic structure calculations and molecular dynamics](#) [8]

**Brief Description**

A general molecular ab initio quantum chemistry package

---

**Source URL:** <https://www.westgrid.ca/support/software/gamess>