## GAMESS 2023R2 (2023Sep30)

### ウェブページ

https://www.msg.chem.iastate.edu/gamess/index.html

### バージョン

2023R2 (2023Sep30)

### ビルド環境

- Intel oneAPI Compiler Classic 2023.1.0
- HPC-X 2.16 (Open MPI 4.1.5)

### ビルドに必要なファイル

- gamess-current.tar.gz (2023R2)
- gmsnbo.i8.a (NBO7.0)
- rungms_rccs_openmpi3 (インストールされた rungms をご覧下さい)
- exam43.patch

```
--- tests/standard/exam43.inp.orig      2018-03-13 11:58:15.322187865 +0900
+++ tests/standard/exam43.inp   2018-03-13 11:58:32.049289234 +0900
@@ -48,7 +48,7 @@
 ! geometry in $DATA, although this is not necessary.
 !
  $contrl scftyp=rhf runtyp=g3mp2 $end
- $system timlim=2 mwords=2 memddi=5 $end
+ $system timlim=2 mwords=30 memddi=5 $end
  $scf    dirscf=.true. $end
  $data
 Methane...G3(MP2,CCSD(T))
```

- pbs_remsh

```
#!/bin/sh
host="$1"
shift
/usr/bin/ssh -n "$host" env PBS_JOBID="$PBS_JOBID" pbs_attach $*
```

### ビルド手順

```
#!/bin/sh
VERSION=2023R2
DIRNAME=gamess/${VERSION}
INSTDIR=~/gamess/deploytest #/apl/${DIRNAME}

# files and patches
MYROOT="${HOME}/Software/${DIRNAME}"
GAMESS_TARBALL="${MYROOT}/gamess-current.tar.gz"
#echo ${GAMESS_TARBALL}
GAMESS_NBOI8A="${HOME}/Software/gamess/2021Jun30/gmsnbo.i8.a"
PATCH_EXAM43="${MYROOT}/exam43.patch"
RUNGMS_RCCS="${MYROOT}/rungms"
PBS_REMSH="${MYROOT}/pbs_remsh"
PARALLEL=12
#-------------------------------------------------------------------------------
umask 0022
export LANG=C
export LC_ALL=C
module -s purge
module -s load pbs openmpi/4.1.5-hpcx2.16/icc2023.1.0
. ~/intel/oneapi/compiler/2023.1.0/env/vars.sh
```

```
. ~/intel/oneapi/mkl/2023.1.0/env/vars.sh

# Load MPI
OMPIDIR=$HPCX_MPI_DIR  #~/hpcx/hpcx-v2.16/hpcx-rebuild-intel2023.2.0

if [ ! -d ${INSTDIR} ]; then
   mkdir ${INSTDIR}
fi
cd ${INSTDIR}
if [ -d gamess ]; then
   mv gamess gamess-erase
   rm -rf gamess-erase &
fi
tar zxf ${GAMESS_TARBALL}
mv ${INSTDIR}/gamess/* .
rm -rf ${INSTDIR}/gamess # maybe already empty

for f in comp compall config lked gms-files.csh runall ddi/compddi \
             tools/remd.csh tools/localgms tools/libxc/download-libxc.csh \
             tools/mdi/download-mdi.csh; do
   sed -i -e "1s/.*/#\!\/bin\/csh -f/" $f
done

 # #patch -p0 <${PATCH_EXAM43}
 cp ${PBS_REMSH} .

expect << EXPECT
spawn csh -f ./config
expect "After the new window is open"
send "\r"
expect "please enter your target machine name:"
send "linux64\r"
expect "GAMESS directory?"
send "${INSTDIR}\r"
expect "GAMESS build directory?"
send "${INSTDIR}\r"
expect "Version?"
send "\r"
expect "HPC system target for 64-bit Linux system"
send "\r"
expect "Please enter your choice of FORTRAN:"
send "oneapi-ifort\r"
expect "Hit <ENTER> to continue to the math library setup."
send "\r"
expect "'acml', 'atlas', 'libflame', 'mkl', 'openblas', 'pgiblas', 'nvblas', 'armpl', 'none'"
send "mkl\r"
expect "MKL pathname?"
send "${MKLROOT}\r"
expect "Hit <ENTER> to continue to the GAMESS DDI communications setup."
send "\r"
expect "communication library ('serial','sockets' or 'mpi' or 'mixed')?"
send "mpi\r"
expect "Enter MPI library"
send "openmpi\r"
expect "Please enter your openmpi"
send "${OMPIDIR}\r"
expect "if you want to accept the default answer of 'no'."
send "\r"
expect "Optional: Build LibXC interface?"
send "yes\r"
expect "Hit <ENTER>"
send "\r"
expect "Optional: Build MDI support?"
send "yes\r"
expect "Hit <ENTER>"
```

```
send "\r"
expect "Optional: Build MSU"
send "yes\r"
expect "Do you want to use LIBCCHEM 2.0?"
send "no\r"
expect "Do you want to use LIBCCHEM"
send "no\r"
expect "Build GAMESS with OpenMP thread support?"
send "yes\r"
expect "Optional: Build GAMESS with VeraChem's VM2 library? (yes/no):"
send "no\r"
expect "Optional: Build GAMESS with TINKER plug-in? (yes/no):"
send "no\r"
expect "Optional: Build GAMESS with VB2000 plug-in? (yes/no):"
send "yes\r"
expect "Optional: Build GAMESS with XMVB plug-in? (yes/no):"
send "no\r"
expect "Optional: Build GAMESS with NEO plug-in? (yes/no):"
send "yes\r"
expect "Optional: Build GAMESS with NBO plug-in? (yes/no):"
send "yes\r"
expect "choice of i4 or i8 integers):"
send "${GAMESS_NBOI8A}\r"
expect "Optional: Build GAMESS with RISM-SCF-cSED plug-in? (yes/no):"
send "no\r"
expect eof
EXPECT

sed -i -e "s/MDI_INSTALL/GMS_3RD_PATH/" Makefile
sed -i -e "s/mdi.mod/mdi*.mod/" Makefile
make ddi

csh -f ./tools/libxc/download-libxc.csh
make -j ${PARALLEL} libxc
csh -f ./tools/mdi/download-mdi.csh
make -j ${PARALLEL} libmdi

/bin/cp -f 3rd-party/mdi/objdir/MDI_Library/*.mod 3rd-party/include/mdi
/bin/cp -f 3rd-party/mdi/objdir/MDI_Library/*.mod object/
make modules

make -j ${PARALLEL}

mv rungms rungms.orig
cp ${RUNGMS_RCCS} ./rungms
chmod -R o-rwx source object libcchem
chmod -R o-rwx ddi/src ddi/server ddi/kickoff
find . -name "src" | xargs  chmod -R o-rwx

export GMSPATH=${INSTDIR}
sed -i -e "s/tools\/localgms/rungms/"        -e "/RUNGMS/s/VERSION)/VERSION) \$(NCPUS)/"        Makefile
export OMP_NUM_THREADS=8
make checktest
make clean_exams
export OMP_NUM_THREADS=1
TEST_LIST="eda qmefpea efp-ci standard"
for tp in $TEST_LIST; do
 time make checktest NCPUS=8 EXAM_PATH=$tp
done

ipcrm -a
```