

GAMESS 2024R2 (2024Jul15)

Webpage

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

Version

2024R2 (2024Jul15)

Build Environment

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

Files Required

- gamess-current.tar.gz (2023R2)
- gmsnbo.i8.a (NBO7.0)
- rungms (please check installed rungms)
- pbs_remsh

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

Build Procedure

```
#!/bin/sh
VERSION=2024R2
DIRNAME=games/${VERSION}
INSTDIR=/apl/${DIRNAME}

# files and patches
MYROOT="${HOME}/Software/${DIRNAME}"
GAMESS_TARBALL="${MYROOT}/games-current.tar.gz"
GAMESS_NBOI8A="${HOME}/Software/games/2021Jun30/gmsnbo.i8.a"
RUNGMS_RCCS="/apl/games/2022R2/rungms"
PBS_REMSH="${MYROOT}/pbs_remsh"
PARALLEL=12
#-----

log_message() {
    echo "[$(date '+%Y-%m-%d %H:%M:%S')] $1"
}

cleanup_ipc() {
    log_message "Cleaning up IPC resources..."
    for type in m s q; do
        ipcs -$type | grep $USER | awk '{print $2}' | while read -r id; do
            if ! ipcrm -$type "$id" 2>/dev/null; then
                log_message "Warning: Failed to remove IPC resource (type: $type, id: $id)"
            fi
        done
    done
}

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 mkl/2023.2.0
. ~/intel/oneapi/compiler/2023.1.0/env/vars.sh
```

```

# Load MPI
OMPIIDIR=$HPCX_MPI_DIR

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

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 "${MKLROOT}\r"
expect "Hit <ENTER> to continue to the GAMESS DDI communications setup."
send "\r"
expect "communication library ('serial','sockets' or 'mpi' or 'mixed')?"
send "mpi\r"
expect "Enter MPI library"
send "openmpi\r"
expect "Please enter your openmpi"
send "${OMPIIDIR}\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 "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 ${RUNGMS_RCCS} ./rungms
sed -i 's|^set GMSPATH=.*|set GMSPATH=${INSTDIR}|' rungms
sed -i 's|^set \s*DDI_MPI_ROOT\s*=.*|set DDI_MPI_ROOT = "${MPI_HOME}"|' rungms
chmod -R o-rwx source object #libcchem is removed
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

cleanup_ipc

```