

## Siesta 4.0.2 for LX

### ウェブページ

<https://launchpad.net/siesta>

### バージョン

#### 4.0.2

### ビルド環境

- Intel Parallel Studio 2017 Update 8

### ビルドに必要なファイル

- siesta-4.0.2.tar.gz
- netcdf-c-4.6.3.tar.gz
- netcdf-fortran-4.4.4.tar.gz
- arch.make (Src/configure を利用することもできます)

```
.SUFFIXES:
.SUFFIXES: .f .F .o .a .f90 .F90

SIESTA_ARCH=rccs-intel17-mkl

FC=mpiifort
FC_SERIAL=ifort
FPP=${FC} -E -P
FFLAGS= -O2 -ip -xHost -fp-model source
RANLIB=ranlib
SYS=nag
LDFLAGS=
FCFLAGS_fixed_f=-fixed
FCFLAGS_free_f90=-free
FPPFLAGS_fixed_F=-fixed
FPPFLAGS_free_F90=-free
BLAS_LIBS=-mkl=sequential
LAPACK_LIBS=-mkl=sequential
BLACS_LIBS=-L$(MKLRROOT)/lib/intel64 -lmkl_blacs_intelmpi_lp64 -lmkl_intel_lp64 -lmkl_sequential -lmkl_core -lpthread -lm -ldl
SCALAPACK_LIBS=-L$(MKLRROOT)/lib/intel64 -lmkl_scalapack_lp64 -lmkl_blacs_intelmpi_lp64 -lmkl_intel_lp64 -lmkl_sequential -lmkl_core -lpthread -lm -ldl
MPI_LIBS=
COMP_LIBS=
NETCDF_ROOT=$(INSTDIR)/netcdf
NETCDF_INCFLAGS=-I$(NETCDF_ROOT)/include
NETCDF_LIBS=$(NETCDF_ROOT)/lib/libnetcdf.a \
    $(NETCDF_ROOT)/lib/libnetcdf.a -lcurl -lhdf5 -lhdf5_hl
MPI_INTERFACE=libmpi_f90.a
MPI_INCLUDE=.
FPPFLAGS= -DMPI -DCDF -DFC_HAVE_ABORT -DFC_HAVE_FLUSH
LIBS=$(NETCDF_LIBS) $(SCALAPACK_LIBS) $(BLACS_LIBS) $(LAPACK_LIBS) \
    $(MPI_LIBS) $(COMP_LIBS)
FFLAGS_DEBUG=-g

#atom.o: atom.f
# $(FC) -c $(FFLAGS_DEBUG) $<
#electrostatic.o: electrostatic.f
# $(FC) -c $(FFLAGS_DEBUG) $<

.F.o:
$(FC) -c $(FFLAGS) $(INCFLAGS) $(FPPFLAGS) $(FPPFLAGS_fixed_F) $<
.F90.o:
$(FC) -c $(FFLAGS) $(INCFLAGS) $(FPPFLAGS) $(FPPFLAGS_free_F90) $<
.f.o:
```

```
$(FC) -c $(FFLAGS) $(INCFLAGS) $(FCFLAGS_fixed_f) $<
.f90.o:
$(FC) -c $(FFLAGS) $(INCFLAGS) $(FCFLAGS_free_f90) $<
```

## ビルド手順

```
#!/bin/sh

VERSION=4.0.2
INSTDIR=/local/apl/lx/siesta402
WORKDIR=/work/users/${USER}
BASEDIR=/home/users/${USER}/Software/Siesta/${VERSION}
TARBALL=${BASEDIR}/siesta-${VERSION}.tar.gz
ARCHMAKE=${BASEDIR}/arch.make

NETCDF_C_VERSION=4.6.3
NETCDF_F_VERSION=4.4.4
BASEDIR_NETCDF=/home/users/${USER}/Software/NETCDF
TARBALL_NETCDF_C=${BASEDIR_NETCDF}/c${NETCDF_C_VERSION}/netcdf-c-${NETCDF_C_VERSION}.tar.gz
TARBALL_NETCDF_F=${BASEDIR_NETCDF}/f${NETCDF_F_VERSION}/netcdf-fortran-${NETCDF_F_VERSION}.tar.gz

PARALLEL=12 # NOTE: parallel make cannot be used for siesta

#-----

module purge
module load intel_parallelstudio/2017update8

export LANG=C
export LC_ALL=C
export FC=ifort
export CC=icc

cd ${WORKDIR}
if [ -d netcdf-c-${NETCDF_C_VERSION} ]; then
  mv netcdf-c-${NETCDF_C_VERSION} netcdf-c-erase
  rm -rf netcdf-c-erase
fi
tar zxvf ${TARBALL_NETCDF_C}
cd netcdf-c-${NETCDF_C_VERSION}

./configure --prefix=${INSTDIR}/netcdf
make -j${PARALLEL}
# make -j${PARALLEL} check # can pass correctly
make install

cd ${WORKDIR}
if [ -d netcdf-fortran-${NETCDF_F_VERSION} ]; then
  mv netcdf-fortran-${NETCDF_F_VERSION} netcdf-fortran-erase
  rm -rf netcdf-fortran-erase
fi
tar zxvf ${TARBALL_NETCDF_F}
cd netcdf-fortran-${NETCDF_F_VERSION}

./configure --prefix=${INSTDIR}/netcdf
make -j${PARALLEL}
### nf03_test & f03tst_open_mem fail due to hdf5 problem?
# make LDFLAGS="-L${INSTDIR}/netcdf/lib" -j${PARALLEL} check
make install

cd ${INSTDIR}
if [ -d siesta-${VERSION} ]; then
  mv siesta-${VERSION} siesta-erase
  rm -rf siesta-erase
fi
```

```

tar zxvf ${TARBALL}
mv siesta-${VERSION}/* .
rmdir siesta-${VERSION}

# hidoiyo...
echo >> Tests/OMM_h2o/OMM_h2o.fdf
echo >> Tests/OMM_si/OMM_si.fdf

mkdir bin # install dir

cd Obj
../Src/obj_setup.sh
cp ${ARCHMAKE} .

# build transiesta and run example
make transiesta && mv transiesta ${INSTDIR}/bin
cd ${INSTDIR}/Examples/TranSiesta/Elec && mkdir OUT_Test && cd OUT_Test
cp ../*.
${INSTDIR}/bin/transiesta < elec.fast.fdf > elec.fast.out
cd ../../Scat && mkdir OUT_TS_Test && cd OUT_TS_Test
cp ../*.
cp ../../Elec/OUT_Test/elec.fast.TSHS .
${INSTDIR}/bin/transiesta < scat.fast.fdf > scat.fast.out

# build siesta
cd ${INSTDIR}/Obj
make clean && make && mv siesta ${INSTDIR}/bin

# tests
cp -r Tests Tests.serial
cd Tests
make SIESTA="mpirun -np 4 ${INSTDIR}/bin/siesta" check >& make_check.log
cd ../ && mv Tests Tests.mpi && mv Tests.serial Tests && cd Tests
make SIESTA="${INSTDIR}/bin/siesta" check >& make_check.log

```

## メモ

- バイナリは /local/apl/lx/siesta402/bin 以下にあります。
  - module 名は siesta/4.0.2 です。
- ジョブスクリプトのサンプルは /local/apl/lx/siesta402/samples 以下にあります。
- 以前のバージョンとは異なり、GPLv3 ライセンスで公開されています。
- transiesta もビルドしています。サンプルが実行できることまでは確認しています。
- 公式情報として ifort 使用時は -O2 が推奨されていたため、-O2 でビルドしています。
- intel18 + intelmpi(psxe18u2, psxe18u4) を用いると MPI 版がうまく動かなかったため、intel17 を使っています。
  - (OpenMPI については試していません)
- Tests/OMM\_h2o/OMM\_h2o.fdf, Tests/OMM\_si/OMM\_si.fdf はなぜかファイル末尾に改行が無かったので無理矢理追加
  - (少なくとも、インテルコンパイラ利用時はファイル末尾改行を追加しないと正常に読み込めないようです)
- (2019/10/3 更新) 実行時も intel2017 系のライブラリを使うように変更。