

## GENESIS 1.6.0

### Webpage

<https://www.r-ccs.riken.jp/labs/cbrt/>

### Version

1.6.0

### Build Environment

- Intel Parallel Studio XE 2018 Update 4

### Files Required

- genesis-1.6.0.tar.bz2
- tests-1.6.0.tar.bz2
- fortdep.py.patch

```
--- fortdep.py.org    2020-12-28 14:05:33.000000000 +0900
+++ fortdep.py       2020-12-28 14:39:43.000000000 +0900
@@ -24,6 +24,7 @@
 re_fcomment = re.compile( "^(^!)*!(.*$)" )
 re_module   = re.compile( "^(.*;+|s*)s*module\s*([^\s,]*)s*", re.I )
 re_use      = re.compile( "^(.*;+|s*)s*use\s*([^\s,]*)s*", re.I )
+ re_include = re.compile( "^(s*)#include\s*([^\s]+)[^\s]" )
 mod_ext     = ".mod"

def __init__( self, fname = "", ext = ".o" ):
@@ -31,6 +32,7 @@
 self.filename = fname
 self.modules  = []
 self.depmods  = []
+ self.includes = []

def setFilename( self, fname, ext = ".o" ):
    if len(fname) == 0:
@@ -54,6 +56,14 @@
    if ( ma ):
        modname = ma.group(2).lower()
        self.appendDependMods( modname )
+    mi = FortranFile.re_include.search( line )
+    if ( mi ):
+        srcname = mi.group(2)
+        if re.search( r".h$", srcname, re.I ):
+            #self.includes.append( srcname )
+            pass
+        else:
+            self.includes.append( re.sub( r"[a-zA-Z0-9]+$", ".o", srcname ) )
    myf.close()
    self.makeUniqList()

@@ -64,8 +74,9 @@
    self.depmods.append( modname + FortranFile.mod_ext )

def makeUniqList( self ):
- self.modules = list(set(self.modules))
- self.depmods = list(set(self.depmods))
+ self.modules = list(set(self.modules))
+ self.depmods = list(set(self.depmods))
+ self.includes = list(set(self.includes))

def getMyModuleFileNames( self ):
    return self.modules
```

```

@@ -87,7 +98,8 @@
    if m.lower() in mods_avail or not mods_avail:
        depmods.append(m)
    ret = ""
-   ret += self.objname + ": " + self.filename + " " + ".join(depmods) + " " + static_deps
+   ret += self.objname + ": " + self.filename + " " + ".join(depmods) + \
+       " " + ".join(self.includes) + " " + static_deps
    if len(self.modules) > 0:
        ret += "\n"
        ret += " ".join(self.modules) + ": " + self.filename + " " + self.objname

```

## Build Procedure

```

#!/bin/sh

VERSION=1.6.0
BASEDIR=/home/users/${USER}/Software/GENESIS/${VERSION}
SRC_TARBALL=${BASEDIR}/genesis-${VERSION}.tar.bz2
TESTS_TARBALL=${BASEDIR}/tests-${VERSION}.tar.bz2
FORTDEP_PATCH=${BASEDIR}/fortdep.py.patch

INSTALLDIR=/local/apl/lx/genesis160

WORKDIR=/work/users/${USER}
BUILDDIR=${WORKDIR}/genesis-${VERSION}
TESTSDIR=${WORKDIR}/tests-${VERSION}

PARALLEL=12
PARALLEL_TESTS=8

# -----
umask 0022

module purge
module load intel_parallelstudio/2018update4

export LANG=C
export LC_ALL=C
export OMP_NUM_THREADS=1

cd ${WORKDIR}
if [ -d genesis-${VERSION} ]; then
    mv genesis-${VERSION} genesis-erase
    rm -rf genesis-erase &
fi

if [ -d tests-${VERSION} ]; then
    mv tests-${VERSION} tests-erase
    rm -rf tests-erase &
fi

tar jxf ${SRC_TARBALL}
tar jxf ${TESTS_TARBALL}

cd ${BUILDDIR}
patch -p0 < ${FORTDEP_PATCH}
FC=mpiifort CC=mpiicc ./configure --prefix=${INSTALLDIR}
make depend

make -j ${PARALLEL} && make install

ATDYN=${INSTALLDIR}/bin/atdyn
SPDYN=${INSTALLDIR}/bin/spdyn

cd ${TESTSDIR}/regression_test

```

```
# atdyn tests
./test.py      "mpirun -np ${PARALLEL_TESTS} $ATDYN"
./test_remd.py "mpirun -np ${PARALLEL_TESTS} $ATDYN"
./test_rpath.py "mpirun -np ${PARALLEL_TESTS} $ATDYN"
./test_gamd.py "mpirun -np ${PARALLEL_TESTS} $ATDYN"
./test_vib.py  "mpirun -np ${PARALLEL_TESTS} $ATDYN"

# spdyn tests
./test.py      "mpirun -np ${PARALLEL_TESTS} $SPDYN"
./test_remd.py "mpirun -np ${PARALLEL_TESTS} $SPDYN"
./test_rpath.py "mpirun -np ${PARALLEL_TESTS} $SPDYN"
./test_gamd.py "mpirun -np ${PARALLEL_TESTS} $SPDYN"
./test_fep.py  "mpirun -np ${PARALLEL_TESTS} $SPDYN"
```

## Notes

- All the tests has passed.
- If Intel MPI 2019 or later version is used, binaries do not run correctly (hang up).