

## Dalton 2020.1

### Webpage

<https://daltonprogram.org/>

### Version

2020.1

### Build Environment

- GCC 13.1.1 (gcc-toolset-13)
- Intel MPI 2021.14
- MKL 2025.0

### Files Required

- (files are downloaded during the build)

### Build Procedure

```
#!/bin/sh

VERSION=2020.1
INSTALL_PREFIX=/apl/dalton/2020.1

BASEDIR=/home/users/${USER}/Software/Dalton/${VERSION}
DALTON_URL=https://gitlab.com/dalton/dalton.git
BRANCH=Dalton${VERSION}
WORKDIR=/gwork/users/${USER}

PARALLEL=24
export LANG=C
export OMP_NUM_THREADS=1

#-----
umask 0022

module -s purge
module -s load gcc-toolset/13
module -s load intelmpi/2021.14
module -s load mkl/2025.0

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

git clone --recursive -b ${BRANCH} ${DALTON_URL}
cd dalton
for f in $(find . -name "*.py*"); do
  sed -i -e "s/env python$/env python3/" $f
done
sed -i -e "s/env python$/env python3/" setup
sed -i -e "s/COMMAND python/COMMAND python3/" cmake/ConfigExternal.cmake

export MATH_ROOT=/apl/oneapi/mkl/2025.0
./setup \
--prefix=${INSTALL_PREFIX} \
--mpi

cd build
make -j ${PARALLEL}
```

```
ctest -j ${PARALLEL} -L dalton
make install
```

## Tests

All the tests have passed successfully.

## Notes

- If Open MPI 4.1.6 was used, the calculation becomes terribly slow. The reason is not clear (not investigated).
- cc2\_r12\_aux\_ch4\_sym\_a2 test failed when OpenBLAS was used instead of MKL. The performance of OpenBLAS version is comparable to MKL version. (Very rough estimation using only the elapsed time of the tests.)
- Gcc13 version shows better performance than gcc8 version.
- Intel compiler version (2022 and 2023 were tested) failed on some of tests. Open MPI version shows worse performance than Intel MPI one. This trend is the same as in the GCC case.