



Guide to Running DL_POLY Classic

What is DL_POLY Classic?

DL_POLY Classic is a general purpose (parallel and serial) molecular dynamics simulation package derived from the package formerly known as DL_POLY_2 (Version 20) which was originally written by W. Smith and T.R. Forester at Daresbury Laboratory to support the CCP5 project 'Computer Simulation of Condensed Phases'.

Please note that DL_POLY Classic is not the same as DL_POLY 4 – the performance of the latter will be markedly superior on large core counts and for large system sizes. However, for small to medium sized systems on moderate core counts, DL_POLY Classic will be competitive and the instructions for using the two codes are very similar, only the performance is different.

Step 1 - Log in

The example used in this guide is configured to run on the Cardiff Skylake *Hawk* cluster. Connect to *hawklogin.cf.ac.uk* with your Supercomputing user credentials using your preferred method (e.g. PuTTY from a Windows machine or ssh from any Linux terminal), thus

```
ssh -l your_username hawklogin.cf.ac.uk
```

The steps below involve typing commands (**in bold font**) in the terminal window.

Step 2 - Load a DL_POLY Classic module

In common with most of the application guides in this series, we assume at the outset that the module of choice would be selected from those originally available on HPC Wales. Thus a necessary first step would be to gain access to that module set by issuing the commands

```
module purge  
module load hpcw
```

Note that the “module purge” command is required to prevent any unintentional collision with pre-existing modules. Subsequently issuing the module command

```
module avail
```

will provide visibility of and access to the entire set of HPCW modules.

More than one DL_POLY Classic package may be available. Note that in common with most other software packages on the system, these are built with the Intel compiler.

- List preinstalled DL_POLY Classic versions:

```
module avail dlpoly-classic
```

- Load the default version:

```
module load dlpoly-classic
```

- Confirm the loaded modules. All dependencies are handled automatically via the module file:

```
module list
```

Step 3 - Create a directory

Create a directory to hold any user data files. For this example, create a directory called DLPOLY-CLASSIC under your home directory:

```
cd ~
mkdir DLPOLY-CLASSIC
```

Step 4 - Obtain a test case

A number of benchmark test cases are provided with the installation at

```
/app/chemistry/dlpoly-classic/1.8/example
```

- Copy the two benchmark files for Bench4 to your user space:

```
cd ~/DLPOLY-CLASSIC
cp /app/chemistry/dlpoly-classic/1.8/example/Bench4.SLURM.q .
cp /app/chemistry/dlpoly-classic/1.8/example/bench4.tar.gz .
tar xzvf bench4.tar.gz
```

The bench4 subdirectory should contain all required files to run a DL_POLY Classic job:

- CONFIG contains the initial positions and velocities for the ions,
- FIELD defines the force-field and,
- CONTROL holds the control parameters for the job (e.g. number of steps, temperature, etc.). The directory also contains sample output files and a batch script.

This particular 80-core benchmark is a simulation of a sodium chloride melt with Ewald sum electrostatics. The system is comprised of 27,000 atoms.

Step 5 - Submit a job

Now you are ready to run this test case with the supplied job script.

- From your working directory, submit the job by:

```
sbatch Bench4.SLURM.q
```

- Check the job queue using:

```
squeue
```

- When completed, output can be found in the file bench4.out.Hawk.80.<Job_ID> in the DLPOLY-CLASSIC subdirectory (where <Job_ID> is the ID assigned by the SLURM queuing system). This can be compared to the reference results in the file

```
/app/chemistry/dlpoly-classic/1.8/example/bench4.out.Hawk.80.15473
```

Step 6 - More test cases

More test cases are distributed with DL_POLY Classic. They can be found at `/app/chemistry/dlpoly-classic/1.8/example`

1. **Bench5** is a simulation of Na/K Disilicate Glass with 3 body forces
 - Copy the two benchmark files for Bench5 to your user space:

```
cd ~/DLPOLY-CLASSIC
cp /app/chemistry/dlpoly-classic/1.8/example/Bench5.SLURM.q .
cp /app/chemistry/dlpoly-classic/1.8/example/bench5.tar.gz .
tar xzvf bench5.tar.gz
```
 - From your working directory, submit the job by:

```
sbatch Bench5.SLURM.q
```
2. **Bench7** is a simulation of Gramicidin A with TIP3P Water solvating
 - Copy the two benchmark files for Bench7 to your user space:

```
cd ~/DLPOLY-CLASSIC
cp /app/chemistry/dlpoly-classic/1.8/example/Bench7.SLURM.q .
cp /app/chemistry/dlpoly-classic/1.8/example/bench7.tar.gz .
tar xzvf bench7.tar.gz
```
 - From your working directory, submit the job by:

```
sbatch Bench7.SLURM.q
```

The generated outputs from both Bench5 and Bench7 can be compared to the reference results in the files:

`/app/chemistry/dlpoly-classic/1.8/example/bench5.out.Hawk.80.15474`

and

`/app/chemistry/dlpoly-classic/1.8/example/bench7.out.Hawk.80.15475`

References

- Official DL_POLY Classic home page: http://www.ccp5.ac.uk/DL_POLY_C/
- CCPForge distribution site: http://ccpforge.cse.rl.ac.uk/gf/project/dl_poly_classic/
- User manual: https://www.ccp5.ac.uk/sites/www.ccp5.ac.uk/files/dl_poly_classic/USRMAN.pdf