Quick reference sheet

Analysing with ONE view

```
$ maqao oneview --create-report=one -options=…
```

Main options:

- `--binary=bin_path`: Path to application executable. Can be relative.
- `--run_command=run_cmd`: Command to run the application, using keyword `<binary>` to reference `bin_path`. If omitted, considered to be `./<binary>`
- `-xp=exp_dir`: Path to directory storing the results. If omitted, directory `maqao_<timestamp>` will be created in the current directory.
- `--format=out_format`: Output format. Accepted values are `html` (default), `xlsx`, `text` and `all` (for all three formats).

Parallel options:

- `--omp_num_threads=num`: Number of OpenMP threads. Will initialise the OMP_NUM_THREADS environment variable.
- `--mpi_command=mpi_cmd`: MPI runtime invocation. Will prepend `run_cmd`.

Batch scheduling options:

- `--batch_script=script_path`: Path to job scheduler script. The script must have been modified to replace the application executable and its arguments with keyword `<run_command>`.
- `--batch_command=batch_cmd`: Command for invoking the job scheduler, using keyword `<batch_script>` to reference `script_path`.

Using a configuration script for ONE View:

- `--config=config_path`: Uses script `config_path` to retrieve options. Options in `config_path` are identical to those from command line (without the --) and declared as Lua variables (`option="value"` or `option=number`).
- `--create-config=sample_config`: Generates a sample configuration script. If `sample_config` is omitted, file `config.lua` will be created in the current directory.

Viewing results:

- Text results are displayed directly on the console output.
- HTML and XLSX results are generated in directory `<exp_dir>/RESULTS/`
  - HTML results are in subdirectory `<binary_name>_one_html`. Open file `index.html` in this directory to display them in your browser.
  - XLSX results are in file <binary_name>_one_0_0.xlsx
- The path to the HTML/XLSX results are displayed at the end of ONE View analysis.

Sample invocations of ONE View

- Command line on interactive MPI run

```
$ maqao oneview --create-report=one --binary=my_path/my_app \
--mpi_command="mpirun -n 4" --omp_num_threads=4
```

- Command line for job scheduler script (script must be edited to replace `my_path/my_app -arg=foo` with `<run_command>`)

```
$ maqao oneview --create-report=one --binary=my_path/my_app \
--run_command="<binary> -arg=foo" \
--batch_script="my_script.job" \
--batch_command="my_jobsched <batch_script>"
```

- Using ONE View configuration script

```
$ maqao oneview --create-config=my_config.lua
{edit my_config.lua to fill useful variables}
$ maqao oneview --create-report=one --config=my_config.lua
```

Advanced: Invoking LProf / CQA separately

Profiling with MAQAO LProf

- Sequential / OpenMP profiling

If ***exp_dir*** is omitted, a directory named ***maqao_lprof_<timestamp>*** will be created.

```
$ maqao lprof [-xp=exp_dir] -- ./foo arg1 arg2 …
```

- MPI / hybrid profiling

```
$ maqao lprof [-xp=exp_dir] -mpi-command="mpirun -n 32" \
-- ./foo arg
```

- Displaying profiling results

```
$ maqao lprof -xp=exp_dir -df # Functions profiling results
$ maqao lprof -xp=exp_dir -dl # Loops profiling results
```

Analysis with CQA

- Analysing a given loop or set of loops

id1, *id2*, *id3* … are the numerical loop identifiers returned by **LProf**.

```
$ maqao cqa ./my_app -loop=id1,id2,id3…
```

- Analysing all innermost loops in a given function or set of functions

```
$ maqao cqa ./my_app -fct-loops="regexp"
```

- Analysing the body of a given function or set of functions

regexp is a regular expression: *foo* matches “foo1”, “foo” or “afoo”, while ^*bar$* matches “bar” only

```
$ maqao cqa ./my_app fct-body="regexp"
```