runxtb.bash – usage and options

Usage

The runxtb wrapper is intended to be an interface to the actual xtb program. Apart from providing a couple of more options to set the environment variables, there should not be any difference in usage.

If runxtb is linked (or found) from a directory along the `PATH` environment variable, (the configuration script prompts to create a symbolic link in `~/bin`,) you can call the script as if you would call the original program:

`runxtb.sh [script options] <coord_file> [xtb options]`

Any provided options to runxtb will overwrite the corresponding rc settings, which in turn take precedence over the built-in defaults. Therefore, if the same options or modes are specified multiple times, only the last one will have an effect, e.g. specifying `-sSi` will run interactively (immediately).

Available options

Processors

Command line: `-p <ARG>`
Configuration: `requested_numCPU=<ARG>`

Specify the number of processors to be used. This will set the environment variables `OMP_NUM_THREADS=<ARG>` and `MKL_NUM_THREADS=<ARG>`. The default and fallback in the script is `4`.

Memory

Command line: `-m <ARG>`
Configuration: `requested_memory=<ARG>`

Specify the memory to be used (in megabytes). This will set the environment variable `OMP_STACKSIZE=<ARG>`. The default and fallback in the script is `1000`.

Wall time

Command line: `-w <ARG>`
Configuration: `requested_walltime="24:00:00"`

Specify the wall time to be used (`[[HH]:[MM]:]SS`). (Default: `24:00:00`)

Output

Command line: `-o <ARG>`
Configuration: `output_file='',0,auto|stdout,-`

The output (without errors) of `xtb` can be trapped into a file called specified.
The default behaviour in non-interactive mode is to guess the filename: If the first argument given after the script options is a readable file, i.e. a `coord_file` or `coords.xyz`, the job- and output-name will be based on that. If the executed program is not `xtb` (`-C` switch, see below), its guess will be based on that. As fallback it will be derived from the current working directory.
If necessary, the automatic generation of the file name can be also be triggered with `-o ''` (the space is important), `-o0`, or `-o auto`. This may be useful to overwrite any previous settings. To send the output stream to standard output, e.g. the terminal, settings can be overwritten with `-c stdout`, or `-c -`. the default mode is `auto`.

Submission modes

Command line: `-i`, `-s`, `-S` Configuration: `run_interactive=yes|sub|no`

The default behaviour of runxtb is to be executed interactively (`-i`). As another feature it is possible to write a bash submission script for a queuing system (see below). This may be useful to make additions to this script, or reviewing it before submission (`-s`). The created job script can also be directly submitted to a specified queue (`-S`).

Queueing system

Command line: `-Q <ARG>`
Configuration: `request_qsys=<ARG>`

In order for the `-s` and `-S` switches to work, a queueing system for which the submit script should be prepared must be set. Currently supported are `pbs-gen` (the default), `slurm-gen`, `slurm-rwth`, `bsub-gen`, and `bsub-rwth` The `*rwth` suffix will test a few more options and will set some constraints according to the recommendations of the RWTH IT centre (this is currently v0.4.0 maintained only as backwards compatibility and will be removed from future versions).

Accounting

Command line: `-P <ARG>`
Configuration: `qsys_project=<ARG>`

In some queueing systems, computational time can be accounted to projects or accounts. When writing a job script, this will include the specified account into the queueing system magic cookies. This is only implemented for the LSF and Slurm queues. When empty (or `default`) a warning will be written to screen.

Path and module settings

Command line: `-M`
Configuration: `use_modules=true|false`

Enable or disable (default) the use of pre-installed modules instead of path settings. This option also needs a specified module or a list of modules (see below).

Command line: `-l <ARG>` Configuration: `load_modules[<N>]=<ARG>`, with `<N>` being the integer load order

Specify a module to be used. This will also invoke `-M`. The option may be specified multiple times to create a list (stored as an array). If `<ARG>` is `0`, the list will be cleared first. The modules (if more than one) need to be specified in the order they have to be loaded if they depend on each other.

Command line: `-B <ARG>`
Configuration: `xtb_install_root="/path/to/xtb/root"`, `xtb_callname="xtb.dummy"`

Set the absolute path to the executable `xtb` to `<ARG>`. The name of the program needs to be included. In the configuration file these are separated. They default to the installation directory and the dummy script.

Command line: `-C <ARG>`
Configuration: `xtb_callname="xtb.dummy"`

Set the name of the program directly. This may be useful to access a different executable from the package, e.g. crest (if installed).

Miscellaneous options

Command line: `-q`
Configuration: `stay_quiet=0|1|2|3`

The logging messages of the script can be suppressed incrementally (`0`), i.e. if specified twice (`2`), it will also suppress warnings, if specified more than twice, it will suppress also errors (`>2`).

Command line: `-h`

Prints a small help text and current configuration.

Command line: `-H`

Retrieve the man page of xtb from the original xtb distribution.

Command line: `-X`

Retrieve the man page of xcontrol from the original xtb distribution.

(Martin; 2020-04-40; wrapper version 0.4.0)