

# AtChem2

## Overview

AtChem2 is a modelling tool for atmospheric chemistry. It is primarily designed to use the Master Chemical Mechanism (MCM), but it can be used with any general set of chemical reactions. The MCM is a near-explicit chemical mechanism which describes the gas-phase oxidation of volatile organic compounds (VOC) in the lower atmosphere. The MCM is available at <http://mcm.york.ac.uk/>. The latest stable version of AtChem2 can be downloaded [from GitHub](#).

This documentation will take you through getting a copy of the AtChem2 source code, setting up the environment for AtChem2 use, building a model, and submitting a model run to Viking's job scheduler, in batch mode.

## Setting Up The Environment

- Load the Viking software modules that you need to work with AtChem2. These are:
  - A Fortran compiler:

### Loading a Fortran Compiler

```
module load toolchain/foss/2021b
```

- git:

### Loading Git

```
module load tools/git/2.33.1-GCCcore-11.2.0-nodocs
```

(git is only needed for downloading AtChem2 source code, not for model compiling or running)

- The SUNDIALS solving suite:

### Loading SUNDIALS

```
module load math/SUNDIALS/2.7.0-foss-2021b
```

- The OpenLibm maths library:

### Loading OpenLibm

```
module load math/OpenLibm/0.8.1-GCC-11.2.0
```

- Clone a copy of the AtChem2 source code:

### Cloning the AtChem2 Repository

```
git clone https://github.com/AtChem/AtChem2.git atchem2
```

- Change directory into AtChem2:

### Moving Into AtChem2

```
cd atchem2
```

At this point, the environment is set up and you are ready to build an AtChem2 model.

## Building a Model

The 'Installation, Setup, and Execution' section of the [AtChem2 README](#) describes the process of building a model, which will be paraphrased here.

1. Make a copy of the AtChem2 Makefile template into a new *Makefile* - this will be used by the model build script to compile the model:

### Copying The AtChem2 Makefile Template

```
$ cp tools/install/Makefile.skel ./Makefile
```

(If you're interested in learning more about what a Makefile is, see [here](#))

2. Open the *Makefile* in a text editor (if you're not sure how to do this, see [here](#)), and change the values of the *CVODELIB* and *OPENLIBMDIR* to correspond with the Viking module environment, i.e.:

### Editing The AtChem2 Makefile

```
CVODELIB      = cvode/lib  
OPENLIBMDIR   = openlibm-0.4.1
```

becomes:

### Editing the AtChem2 Makefile (2)

```
CVODELIB      = /opt/apps/easybuild/software/SUNDIALS/2.7.0-foss-2021b/lib  
OPENLIBMDIR   = /opt/apps/easybuild/software/OpenLibm/0.8.1-GCC-11.2.0/lib
```

3. Run the AtChem2 build script, giving it the location of a [FACSIMILE](#) format mechanism with which to build your model (for this example, we are using the test mechanism supplied with AtChem2):

### Running the AtChem2 Build Script

```
$ ./build/build_atchem2.sh mcm/mechanism_test.fac
```

You will see some text describing what this process is doing, as well as some compiler warnings - don't worry about these! If this process completes successfully, you should now have an AtChem2 executable file in the current directory.

#### 4. Test that the mechanism test model built correctly by running it:

##### Testing the Model

```
./atchem2
```

If the model runs successful, you should see a 'final statistics' statement in the output that looks like the following:

##### Checking AtChem2 Output

```
-----  
Final statistics  
-----  
No. steps = 603   No. f-s = 699   No. J-s = 1003   No. LU-s = 100  
No. nonlinear iterations = 696  
No. nonlinear convergence failures = 0  
No. error test failures = 28  
  
Runtime = 0  
Deallocating memory.
```

At this point, you can build and run AtChem2 models, and are ready to start using your own mechanisms and configuring the model parameters for your simulations. Please refer to the [AtChem2 wiki](#) or the [AtChem2 User Manual](#) for comprehensive information about how to work with AtChem2.

## Running a Model

This section will cover the basics needed to get a compiled AtChem2 model running on Viking in batch mode. For a full explanation of Viking job submission, and details on job parameters, see [here](#) and [here](#).

### Writing a Job Script

Importantly, a Slurm job script tells Slurm, the Viking job scheduler, what you want to do, how long you want to do it for, and specifies the resources you will need to do it. Whilst you can provide this information every time you want to run a job, using the command-line options to [sbatch](#), it is convenient to store them in a specially-formatted text file (a 'job script') and tell Slurm to read the information from there instead.

Create a text file in your AtChem2 working directory called *atchem2\_mech\_test.sbatch* (the name and extension of this script are inconsequential!), containing the following:

### Example AtChem2 Job Script

```
#!/usr/bin/env bash

#SBATCH --job-name=atchem2_mech_test
#SBATCH --mail-user=USERNAME@york.ac.uk
#SBATCH --mail-type=ALL
#SBATCH --output=atchem2_mech_test.log
#SBATCH --account=VIKING-PROJECT-ACCOUNT-CODE

#SBATCH --partition=nodes
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=1
#SBATCH --mem-per-cpu=100MB
#SBATCH --time=00:02:00

cd ${SLURM_SUBMIT_DIR} || exit

module load toolchain/foss/2021b
module load math/SUNDIALS/2.7.0-foss-2021b
module load math/OpenLibm/0.8.1-GCC-11.2.0

./atchem2
```

Take care to replace *USERNAME* with your University of York username, and *VIKING\_PROJECT\_ACCOUNT\_CODE* with the Viking project code that your supervisor should have given to you when they signed up for a Viking project.

This script achieves the following:

- Sets up some job metadata, including the name of the job, the email address to which all job information should be sent, the name of the file into which the core job output will be stored (not the model outputs, just the diagnostic text that the model produces when you run it), and the code of the Viking project under which the job will be run
- Sets up the job resource requirements:
  - Job will be run in the *nodes* partition (for detail on available Viking resource partitions, see [here](#))
  - Job consists of 1 *task* (1 thing to be done, run an AtChem2 model)
  - Job task needs 1 core
  - Core assigned to the job task needs 100MB of memory (the test AtChem2 run is small!)
  - The job will take no longer than 2 minutes (the test AtChem2 run is short!)
- Specifies what the job should do:
  - Change directory to the directory that the job was submitted from
  - Load the modules that are needed to run AtChem2
  - Run the previously built AtChem2 model

With this job script in place, you are ready to run the model.

### Submitting a Batch Job

With a job script written, submitting the job to Slurm is as straightforward as giving the job script to the *sbatch* command:

#### Submitting an AtChem2 Batch Job

```
$ sbatch atchem2_mech_test.sbatch
```

This will pass the job description to Slurm, which will attempt to schedule and run the job. You will receive information about the job starting and ending via email to the address specified in the job script, including whether or not the job completed successfully. Additionally, the job's output log (*atchem2\_mech\_test.log* or the example job above) will contain the job's standard output, as well as some information about how efficiently you used the resources that were allocated to the job.

In the case of your AtChem2 example job, you will find successfully-run model output in the *model/output* directory:

#### Example AtChem2 Model Output

```
$ cd model/output
$ ls
environmentVariables.output  finalModelState.output  lossRates.output          photolysisRates.
output                       productionRates.output  speciesConcentrations.output
errors.output                jacobian.output         mainSolverParameters.output  photolysisRatesParameters.
output  reactionRates
```

## Additional Support

If you require additional support for running AtChem2 on Viking, please submit a support request to <itsupport@york.ac.uk> - your request will be passed to somebody who will be able to help. Questions about the model itself are best discussed with fellow users of AtChem2, your supervisor, or the model developers themselves ([via GitHub](#)).