UKCA Chemistry and Aerosol Tutorials at vn10.9 using Rose & Cylc

N. Luke Abraham\textsuperscript{1,2}, Nicolas Bellouin\textsuperscript{3} \\
& Anja Schmidt\textsuperscript{1,4}

1. Department of Chemistry, University of Cambridge, U.K.
2. National Centre for Atmospheric Science, U.K.
3. Department of Meteorology, University of Reading, U.K.
4. Department of Geography, University of Cambridge, U.K.

https://doi.org/10.17863/CAM.22153
www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_Tutorials_at_vn10.9

November 2017

Acknowledgements
We would like to thank Alex Archibald, Scott Archer-Nicolls, David Wade, Ewa Bednarz, Mohit Dalvi, Paul Cresswell, David Matthews, Ben Johnson, John Pyle, and the team at NCAS Computational Modelling Services (cms.ncas.ac.uk) for their help and support in the creation of these tutorials. This work used the ARCHER UK National Supercomputing Service (www.archer.ac.uk). This work was supported by the NERC Advanced Training Short Courses scheme, grant number NE/P020089/1.
UKCA Chemistry and Aerosol Tutorials at vn10.9

The following tutorials will teach you how to use various aspects of UKCA, and the parts of Rose (the Unified Model's python-based user interface) that are specific to UKCA.

- **Things to know before you start**

General use of the Rose and UKCA:

1. Running existing UKCA Suite
2. Exploring Rose
3. What is STASH?
4. Adding new chemical tracers
5. Adding new emissions
6. Adding new chemical reactions
7. Adding dry deposition of chemical species
8. Adding wet deposition of chemical species
9. Adding new UKCA chemical diagnostics
10. Examining Aerosol Impacts
11. Worked Example: Developing a change for submission to the trunk

These tutorials are an update and expansion of the tutorials which were at UM8.2, UM8.4, and UM10.4.

This tutorial has been funded by the NERC Advanced Training: Short Courses award scheme (http://www.nerc.ac.uk/funding/available/postgrad/advanced/atsc/) and the ACITES Atmospheric Chemistry Modelling Network (http://www.ncas.ac.uk/index.php/en/what-is-acites).

Written by Luke Abraham 2018


This page was last modified on 18 April 2018, at 15:07.
UKCA Chemistry and Aerosol Tutorials: Things to know before you start the vn10.9 practicals

From UKCA

UKCA Chemistry and Aerosol Tutorials at vn10.9

You will need to have the details of your PUMA and ARCHER accounts to hand.

Contents
- 1 Before you begin the UKCA Tutorials
  - 1.1 Accounts
- 2 Developing code for the UM
- 3 UKCA Training Suite
  - 3.1 Additional Requirements (Virtual Machine)
    - 3.1.1 Tutorial Data
    - 3.1.2 Xconv
    - 3.1.3 Iris
  - 3.2 Manual Compiling (ARCHER)
- 4 Scripts

Before you begin the UKCA Tutorials

These tutorials assume that you are familiar with FCM, and training on this can be found here:
http://cms.ncas.ac.uk/wiki/Fcm

The UM FCM tutorial that is linked from this page uses the old user interface (the "UMUI"), so will be less relevant here.

You should also familiarise yourself with Rose & Cylc, and more information on these can be found here:
http://cms.ncas.ac.uk/wiki/RoseCylc

The Met Office provide some on-line UM specific Rose training here:

Also, if you do not have a PUMA or ARCHER account, you can use the Met Office provided Virtual Machine to run the VM, provided here:
https://github.com/metomi/metomi-vms

You should follow the instructions in UMDP X10 (https://code.metoffice.gov.uk/doc/um/latest/umdp.html).

It is possible to run a simple UKCA suite in the VM, but note that this requires at least 6GB of RAM to run successfully.

Accounts

To run these tutorials the minimum you will need is
- an account on the Met Office Science Repository Service (https://code.metoffice.gov.uk)

which will then allow you to run the tutorials in a Virtual Machine.

To run these on ARCHER you will need
- a PUMA account
- an ARCHER account

For information on how to get these, please see the Getting Started with UKCA page.

If you are attending the face-to-face training workshop you will be provided with ARCHER training accounts, although you still need your own PUMA account.

Developing code for the UM

During this tutorial you will be creating tickets on the Met Office Science Repository Service (https://code.metoffice.gov.uk) , and making code changes to UM branches. Please note that this is a live system, used by scientists all over the world to develop their code. However, please don't worry about breaking anything, as there are practices in place to prevent this.

For more information about developing for the UM, please see the Working Practices for UM Development with Rose, FCM and trac (https://code.metoffice.gov.uk/trac/um/wiki/working_practices).

UKCA Training Suite

<table>
<thead>
<tr>
<th>Machine</th>
<th>UM Version/Configuration</th>
<th>Suite ID</th>
<th>Owner</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARCHER</td>
<td>vn10.9 N48L38 Intel Compiler</td>
<td>u-as101</td>
<td>lukeabraham</td>
<td>May be compiled manually (recommended)</td>
</tr>
<tr>
<td>vm</td>
<td>vn10.9 N48L38 GNU Compiler</td>
<td>u-as159</td>
<td>lukeabraham</td>
<td>1x2 decomposition</td>
</tr>
<tr>
<td>XC40</td>
<td>vn10.9 N48L38 GNU Compiler</td>
<td>u-au234</td>
<td>lukeabraham</td>
<td>4x9 decomposition (36-core)</td>
</tr>
</tbody>
</table>
These suites are based on the um_n48_ukca_eg app, which is available in the MetUM trunk (in rose-stem/app/). A ticket giving more details on these (and the code changes) can be found at: um:#3578.

To set-up using the UM with Rose & Cylc correctly, go to


and follow the instructions from Section 1.2 onwards if you are using Linux/Mac (after opening a terminal).

Note: when entering MOSRS password, a rose window will open behind everything all your other windows. You will also need to enter your username here to complete the process.

When you have completed this, you should launch the Rose browser by typing

```
rosie go &
```

and there you should search for training suite and make a copy of it.

When you have copied the suite, you will need to remember to change the username & tic code in suite conf —> Machine Options.

### Additional Requirements (Virtual Machine)

#### Tutorial Data

To be able to do Tutorials 5 and 10 you will need emissions data and python scripts. These files, as well as worked solutions for all the Rose and code changes can be obtained by running the following command from the home/ directory of your VM:

```bash
wget -q --show-progress http://gws-access.ceda.ac.uk/public/ukca/UKCA_Tutorial_vn109.tgz
```

There is 880MB to download, so it might take some time.

Once the download has finished, you can extract the archive by running

```
tar -xzvf UKCA_Tutorial_vn109.tgz
```

This will make a directory called 'tutorial/' containing 1.4GB of files. If you cd into this you can check that the files have not been corrupted by running the following command:

```
md5sum --status -c vn109.md5
```

#### Xconv

You will need to download Xconv (http://cms.ncas.ac.uk/documents/xconv/) (xconv1.93) from here:

- http://cms.ncas.ac.uk/documents/xconv/_downloads/xconv1.93_linux_x86_64.tar.gz

You can download this by

```
wget http://cms.ncas.ac.uk/documents/xconv/_downloads/xconv1.93_linux_x86_64.tar.gz
```

Download it to your $HOME/bin on the VM, cd into this directory, tar -xzvf the tar-ball, and then

```
ln -s xconv1.93 xconv
ln -s xconv1.93 convsh
```

#### Iris

There is an install-iris script provided, but you will need to set-up modules yourself to be able to use it properly. The anaconda install breaks Rose if put in your PATH, however, there is now an alias

```
conda
```

which will open a new terminal with all the anaconda python packages in its $PATH. This will allow you to use Rose in one terminal and Iris in another.

I have found a handy way to use python is to use ipython (https://ipython.org/) with the following arguments

```
ipython --pylab --logfile=ipython-`date +"%Y%m%d-%H%M%S"`.py
```

I have aliased this to

```
pylab
```

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_Tutorials:_Things_to_know_before_you_start_the_vn10.9_practicals
in my .bashrc. The

```
--pylab
```

sets up a MatLab-type environment (numpy, scipy, matplotlib all loaded using standard shortcuts), and

```
--logfile=ipython-`date +"%Y%m%d-%H%M%S"`.py
```

means that all commands are saved to a file of the format

```
ipython-YMMD-HHMMSS.py
```

Manual Compiling (ARCHER)

The UKCA Training Suite has been altered slightly to allow it to be easily manually compiled. This can be selected as an option in `suite conf` → UM hosts. With this set to true, when the suite runs the `fcm_make2` task will enter the submit-failed state. Then, on ARCHER, you should then change directory into

```
$HOME/cylc-run/[SUITE-ID]/log/job/1/fcm_make2/NN
```

and compile the um-atmos and um-recon executables by typing

```
./job
```

This will take around 7-12 minutes to compile. When it has finished successfully you will get the following message in the terminal, just before the prompt:

```
JOB SCRIPT EXITING (TASK SUCCEEDED)
```

You can follow the progress of this process in detail by tail-ing the following file

```
$HOME/cylc-run/[SUITE-ID]/share/fcm_make/fcm-make2.log
```

Once this step has completed, you can send the rest of the suite off by going back to the Gcylc window and right-clicking on the `fcm_make2` task and click Reset State → "succeeded". This will then allow the rest of the tasks to start in sequence.

Note: manually compiling in this way means that you will no longer have the `job.out` or `job.err` files. You should instead look in the `fcm-make2.log` file.

Scripts

There are a number of scripts provided for these tutorials. These can be found on GitHub here:

- https://github.com/theabro/ukca

They are also available on ARCHER.

UKCA Chemistry and Aerosol Tutorials at vn10.9

This page was last modified on 9 March 2018, at 10:41.
UKCA Chemistry and Aerosol vn10.9 Tutorial 1

From UKCA

UKCA Chemistry and Aerosol Tutorials at vn10.9

Before starting these tutorials you should first make sure that you have completed the setup instructions (http://cms.ncas.ac.uk/documents/training/November2016/UM_practicals/getting-setup.html). If you are using Linux or macOS you will only need to complete section 2.2 onwards.

Contents

- 1 Copying and Running an Existing Rose Suite
- 2 Version Control
- 3 Output Directory Structure
  - 3.1 Notes
- 4 Viewing Output
- 5 Checklist

Copying and Running an Existing Rose Suite

<table>
<thead>
<tr>
<th>Machine</th>
<th>UM Version/Configuration</th>
<th>Suite ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARCHER</td>
<td>vn10.9 N48L38 Intel Compiler</td>
<td>u-as101</td>
</tr>
<tr>
<td>vm</td>
<td>vn10.9 N48L38 GNU Compiler</td>
<td>u-as159</td>
</tr>
<tr>
<td>XC40</td>
<td>vn10.9 N48L38 GNU Compiler</td>
<td>u-au234</td>
</tr>
</tbody>
</table>

If you are using PUMA & ARCHER, you will need to login to PUMA, e.g.

```
ash -Y username@puma.nerc.ac.uk
```

If you are using the Met Office Virtual Machine, you will need to login to the VM, e.g.

```
vagrant ash
```

You should then be asked for your SRS password.

Then launch the UM graphical user interface by:

```
rosie go
```

This should then load up a blank interface. Go to Edit → Data source and select u. The go to the search panel and search for the correct base suite. This will then show all the suites that have this suite-id in it’s history or title. You should just select the correctly named suite and not any of the others.

Right-click on the suite and click copy suite. A new box will open asking for the project - it is fine to press Forward here. On the next panel, it's fine to just press OK again as well. You should avoid the checkout suite option, as this will give you a copy of someone else's suite, and it might be possible for you to make changes to it that could affect other users.

The suite will now copy and checkout to your /home/$USER/roses directory, and will also appear when running rosie go.

You should now right-click → edit (or double-click) on this new suite, and click run, which is symbolised by a play symbol (i.e., a large grey arrow-head pointing to the right).

If you are using PUMA/ARCHER, this suite has been configured to manually compile, so you should follow the instructions on how to do that, to be sure that you can compile your suite as quickly as possible. It will take about 7 minutes to compile the code (fcm_make and fcm_make2 tasks), followed by about 3 minutes to run the reconfiguration step (recon), and then 1 minute to run the UM itself (atmos).

If you are using the VM, times will vary depending on the specifications of the host. It could take about 10 minutes to compile the code (fcm_make) for the first time (and about 1-2 minutes when recompiling), followed by about 1 minute to run the reconfiguration step (recon), and then about 12 minutes to run the UM itself (atmos).

When the suite has finished successfully it will then become blank with the message stopped with 'succeeded' in the bottom-left corner.

Version Control

Rose suites are all held under version control, using fcm (http://cms.ncas.ac.uk/wiki/Fcm). When making changes to a suite, you will need to save it before you can run the suite. Once you are happy with the settings, you can also commit these changes back to the repository - to do this change directory to the

---

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_1
and then type

```
fcm commit
```

a text editor will then open, and you should type a short message describing what the changes you have made do. You should then close the editor and type `y` in the terminal. It is recommended that you commit frequently (even on configurations that aren't working) as this protects you against mistakes and accidental deletions etc.

These suites can be viewed on the SRS here: https://code.metoffice.gov.uk/trac/roses-u (password required) (https://code.metoffice.gov.uk/trac/roses-u)

It is recommended that you commit your suites regularly.

### Output Directory Structure

The output directory structure of rose suites are rather complex. The schematic on the right gives a broad overview of the general structure. Everything can be found within the `[SUITE-ID]` (e.g. `u-as101` etc.) directory, which can be found within your `$HOME/cylc-run` directory. On the VM, everything in held in this single place, but things are more complicated when using PUMA & ARCHER, as some files are held on the ARCHER `/home` filesystem, some on the `/work` filesystem, and some files are synced back to PUMA as well.

Within this directory there are several directories, including:

- **log**: a symbolic link to a directory called something like `log.20171214T141332Z`, which contains all the `job.out` (containing output from UM WRITE statements from the `umPrint` subroutine) and `job.err` files etc., as well as the script used to run the job.
- **work**: this contains the directories used when the job actually runs. Real-time output will be sent to files here (held in a `pe_output` directory. Sometimes model output will also be here. On ARCHER this directory exists on the `/work` filesystem, and some files are synced back to PUMA as well.
- **share**: This isn't used by the UKCA Training Suite, but for climate jobs model output will usually be held in this directory. On ARCHER this directory exists on the `/work` filesystem.

Within these directories there will be many sub-directories. Some of these will be named from the `cycle-point` (labelled `[CYCLE]` in the graphic. For the UKCA Training Suite this will be 1, but under normal climate operation this will be a date string, e.g. 19880901T0000Z. The sub-directories will (eventually) be named after the `app` (labelled by `[JOB NAME]` in the table below) that the output is from, e.g. `fcf_make, recon, atmos` etc.
Example output from the UKCA training suite.

Files

<table>
<thead>
<tr>
<th>Most recent job.out files</th>
<th>Processor output (while running)</th>
<th>Output files in 64-bit fieldsfile format</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARCHER: /home/n02/n02/$USER/cylc-run/[SUITE-ID]/log/job/1/[JOB NAME]/NN</td>
<td>ARCHER: /work/n02/n02/$USER/cylc-run/[SUITE-ID]/log/job/1/[JOB NAME]/pe_output</td>
<td>ARCHER: /work/n02/n02/$USER/cylc-run/[SUITE-ID]/log/job/1/atmos</td>
</tr>
<tr>
<td>vm: /home/vagrant/cylc-run/[SUITE-ID]/log/job/1/[JOB NAME]/NN</td>
<td>vm: /home/vagrant/cylc-run/[SUITE-ID]/log/job/1/[JOB NAME]/pe_output</td>
<td>vm: /home/vagrant/cylc-run/[SUITE-ID]/log/job/1/atmos</td>
</tr>
</tbody>
</table>

**Notes**

- The job.out files can also be viewed through the Gcylc GUI right-click menu from each job.
- A handy command to check the progress of the atmos job is
  ```
  ARCHER: tail -1000f atmos.fort6.pe00 | grep Atm Step
  vm: tail -1000f atmos.fort6.pe0 | grep Atm Step
  ```
- Other suites may put the output fieldsfiles into a different location, e.g.
  ```
  /work/n02/n02/$USER/cylc-run/[SUITE-ID]/share/data/History_Data
  ```
  and there are also post-processing settings that can be used to copy data to the RDF or JASMIN as the suite runs.
- Output fieldsfiles can have various naming conventions. For these suites, they will be `atmosa.pa19810901_00`, but more generally they are likely to be of the form
  ```
  [SHORT SUITE-ID]a.pabcdefghijkmsyxYYYYMDD
  e.g. ag308a.pk19880901
  [SHORT SUITE-ID]a.pabcdefghijkYYYYmon
  e.g. ak468a.pe1989oct
  ```
- Climate-mean files (.pm) will often also have seasonal (.ps), annual (.py), and decadal (.px) equivalents.

**Whilst this suite is running, take a look at Tutorial 2: Exploring Rose.**

**Viewing Output**

For more detailed plotting, tools such as the Iris (http://scitools.org.uk/iris/) and cf-python (http://cfpython.bitbucket.org/) libraries can be used to view UM file formats directly. However, for quick viewing, Xconv (http://cms.ncas.ac.uk/documents/xconv/) is a very useful tool. Information on how to install Xconv on the VM can be found here.

To view these files, do

```
xconv atmosa.pa19810901_00
```

As well as viewing files, you can use Xconv to convert these files to netCDF, by filling in the **Output file name:** box (e.g. `foo.nc`), and then clicking **convert**. If no path is defined, this will save the file in the same directory that you opened Xconv from.

Example output from the UKCA training suite can be found at `/work/n02/n02/ukca/Tutorial/vn10.9/sample_output/Base/atmosa.pa19810901_00` on ARCHER.

**Checklist**

- [ ] List suites using rosie go
- [ ] Copy suites using the right-click menu
- [ ] Run suites using the play button

**Tutorial 2**

Written by Luke Abraham 2017

Retrieved from "http://www.ukca.ac.uk/wiki/index.php?title=UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_1&oldid=6755"

This page was last modified on 18 January 2018, at 12:20.
UKCA Chemistry and Aerosol vn10.9 Tutorial 2

From UKCA

UKCA Chemistry and Aerosol Tutorials at vn10.9

Contents

- 1 Exploring Rose
  - 1.1 fcm_make
  - 1.2 um
    - 1.2.1 STASH Requests
    - 1.2.2 Section 34 - UKCA: UK Aerosols and Chemistry
  - 1.3 search
- 2 Checklist

Exploring Rose

If you are familiar with the UMUI, you will see that Rose is quite different. As well as a complete re-write of the user interface, a lot of the namelists have been streamlined and also other things, such as how STASHmaster files are read (for user diagnostics etc.), have also been changed.

Take a look at your suite, and click through the options. There are 2 key sections that you will need to consider:

fcm_make

Also called often called fcm_make_um.

This section controls which fcm branches are used by the model, and also what compiler settings are used etc. You should go to fcm_make env Sources to see what branches are being used (although there are none used by the UKCA training suite). During the course of these tutorials you will make a branch and add it here to complete the tutorial tasks. You can add both relative paths to the repository (starting with branches/) and absolute paths to a working copy (starting with /home/).

um

Click the arrow by um to view the list of sections. We'll be most interested in the namelist section, which contains all the settings that configure the science that will be done in this configuration of the UM, including the UKCA-specific options, and also when adding more output. We will mostly be concerned with 2 sections.

STASH Requests

Go to um namelist Model Input and Output STASH Requests and Profiles STASH Requests to view all the model output that has been requested by this suite. This list can be quite extensive, and we will cover adding more output in Tutorial 3: What is STASH? During these tutorials this will be know as the STASH panel.

Section 34 - UKCA: UK Aerosols and Chemistry

Go to um namelist UM Science Settings. This contains all the different science sections, and by clicking on each panel you can see the settings for each. If you click on Section 34 - UKCA: UK Aerosols and Chemistry you will be able to see the UKCA settings. During these tutorials this will be known as the UKCA panel.

In Rose there is a 1:1 correspondence between the name of the variable in Rose, and the name of the variable in the namelist and hence the UM. This means that it can be easy to find what variable you might be interested in by first finding the appropriate variable in Rose (e.g. l_ukca_ageair). By the name of each variable is a cog - if you click this you can view info about the variable, which lists all the information the Rose has been given about it, e.g.

```
Data
name_i_ageair_reset_method
value 1
comments
error
flags
ignored_reason
warning

Metadata
compulsory true
description Method of resetting near-surface values of Age-of-air tracer
full ns /um/namelist/UM Science Settings/section34
help There are two options available for resetting the near-surface values of the age-of-air tracer to 0.0:
1. based on maximum level upto which to reset
2. based on maximum height upto which to reset
if 'by level' selected,
recommended value is 10 for consistency
with older UKCA configurations
if 'by height' selected,
```
The standard value is 2000.0 metres

```
! id namelist:run_ukca=i_ageair_reset_method
! sort-key a09
! trigger namelist:run_ukca=max_ageair_reset_level: 1 ;
! namelist:run_ukca=max_ageair_reset_height: 2 ;
! value-titles Reset by level, Reset by height
! values 1,2
```

You can also click help, which lists the help text associated with each variable. If you are not sure about a variable, looking at the info can be very informative. Due to the way the change process of the UM works, all variables must have information, such as the above, supplied, and any new science that has been added also needs to be added to each sections documentation paper.

**search**

Unlike the UMUI, there is a search box in Rose. It works best if you know the name of a variable, although it is still very useful to find options if you're not sure where they are.

**Please take some time to browse through the rose suite, and get a feel for where things are.** If you previously used the UMUI, you might want to try to locate comparable panels to see how things have changed.

**Checklist**

- ☐ Make use of search to find what you need in the suite.

---

Tutorial 3

*Written by Luke Abraham 2017*


- This page was last modified on 3 January 2018, at 11:27.
UKCA Chemistry and Aerosol vn10.9 Tutorial 3

From UKCA
UKCA Chemistry and Aerosol Tutorials at vn10.9

When using the UM, it is likely that you will want to add new diagnostics, and we will cover writing out chemical fluxes in a later tutorial. Here we will cover selecting new diagnostics in STASH.

Contents

1 Task 3.1: add new output
2 What is STASH?
   2.1 Prognostic and Diagnostic Fields
   2.2 STASH Sections and Items
3 STASH Panel
4 UKCA STASH sections
5 Solution to Task 3.1
6 Checklist

Task 3.1: add new output

**TASK 3.1:** Output 3-hourly-mean UKCA CO (section 34 item 010) field to the UPA output stream, and then add-in the equivalent diagnostic on pressure levels using the DP27CCM domain.

**Hint**
You will need to remember to run the TidyStashTransform macro once you have added the diagnostics in the STASH panel.

In the following page you will learn more about how the Unified Model's output is organised and managed. You should use this information to complete the task above. A worked solution is provided at the end of the tutorial.

What is STASH?

STASH is the Unified Model's storage handling and diagnostic system. It is designed to cope with the many different configurations that the UM can be used in, but still provide output in a consistent and standard way. A full technical description of STASH can be found in the Met Office Science Repository Service ([Unified Model Documentation Paper C4](https://code.metoffice.gov.uk/doc/um/latest/um-training/stashmaster.html)) which can be downloaded from the Met Office Science Repository Service (password required). You may also want to look through the UM STASH help at: [https://code.metoffice.gov.uk/doc/um/latest/um-training/stashmaster.html](https://code.metoffice.gov.uk/doc/um/latest/um-training/stashmaster.html).

Prognostic and Diagnostic Fields

The UM considers variables (or fields) to be of two different types, *prognostic* or *diagnostic*. Prognostic fields are ones which the model must have values for, prior to each timestep, as e.g. the equations of motion the model solves require these fields (these are fields such as specific humidity or potential temperature), so they must exist in the model start dump. Diagnostic fields are all other fields that are derived from prognostic ones, and as such the model does not need to have prior values for these. Ancillary files (such as emissions, SSTs etc) contain prognostic fields. UKCA requires around 137 prognostic fields in the GA7+StratTrop configuration, and could in theory provide almost 2000 diagnostic fields on model levels, many of which have pressure-level equivalents.

From a user's perspective, STASH is used to output fields during the run, and from this point of view it does not matter if these are prognostic or diagnostic fields. However, you will need to consider these differences when you add new chemical tracers.

STASH Sections and Items

Each field that is considered by STASH has a unique address which is given by a *section* and an *item* number. Prognostic fields are mostly held in section 0 (with the exception of tracers) and diagnostics are organised by areas of the code, e.g. short-wave radiation diagnostics are held in section 1, long-wave radiation diagnostics are held in section 2 etc. Some sections will always be on, and some sections will only be on if a certain process is selected, e.g. the interactive land-surface scheme. Each section can hold up to 999 items, where each item is a separate prognostic or diagnostic field, and can be either 2D or 3D.

Each field has its own entry in a STASHmaster file. There is a master list of fields which is held in the STASHmaster_A file, a copy of which is located in each branch at:

```
rose-meta/um-atmos/vnX.Y/etc/stash/STASHmaster/STASHmaster_A
```

If you make changes to this, to add your own diagnostics, you will need to make changes to the

```
rose-meta/um-atmos/HEAD/etc/stash/STASHmaster/STASHmaster_A
```

file, and then change where Rose looks for the metadata associated with your suite. This will be covered in more detail when you add new chemical tracers.

The term HEAD here refers to fact that this is the latest metadata of the branch (c.f. the "head" of an FCM branch being the latest version of it). For any new branch at a particular model version (e.g. vn10.9) the STASHmaster_A file in HEAD and e.g. vn10.9 will (initially) be the same. When the branch (at e.g. vnX.Y) is committed to the trunk the changes in HEAD are then merged into the trunk to become part of the vnX.Y+0.1 metadata.

STASH Panel

The STASH panel can be found at um → namelist → Model Input and Output → STASH Requests and Profiles → STASH Requests.
It is usually initially sorted by section/item numbers, but you can change the ordering by clicking on the items on the bar at the top (e.g. use_name). This can be helpful to see which items are going to which output stream.

To add new output, click the New button (with the large plus symbol on it), and this will open the Add new STASH requests panel. This lists all the available STASH output by section. You can scroll down to the section you want, click the small arrow on the left, scroll down to the diagnostic that you want and either right-click on it and click Add STASH request, or select it and then click the Add button in the top right-hand corner of the panel.

Another option, especially if there is already a STASH request that has similar domain, time, and usage profiles, is to right-click the existing request, and select Added request as namelist: streq(1). You will also find that a red warning triangle has appeared in the rose editor. This will disappear once you fill-in the dom_name, tim_name, and use_name entries for it. This can be done by clicking next to the red X that is in each column and selecting the correct value from the drop-down menu.

Once you have selected entries for each usage field, you will need to run the TidyStashTransform macro. When the STASH entry was created it was given the the namelist value streq(1), with the next as streq(2) etc.. This is a default value, and the correct namelist index needs to be generated for each STASH item (each is unique to the choices made, which also prevents duplicate entries being created). To do this, click the Macros drop-down menu from the top-left of the panel and select the stashindices.TidyStashTransform option. This will process the STASH requests, and any entries where the index doesn't match will be listed. Click OK, and this will then apply the correct index to the request, which will be the 5-digit STASH number followed by a string of seemingly-random letters and numbers, e.g. 34010_c4b8f1ad.

It is also possible to run any Rose macro (including stashindices.TidyStashTransform) by going to the Metadata menu at the top of the Rose GUI.

UKCA STASH sections

UKCA actually has 6 STASH sections - these are:

- **Section 34**: This contains the prognostic UKCA variables on model theta levels, i.e. all the transported species, and several other fields which are required to be in the restart file.
- **Section 37**: This is for UKCA tracer lateral boundary conditions for use in limited area configurations.
- **Section 38**: This section is for GLOMAP-mode diagnostics on model theta levels.
- **Section 50**: This section is for Chemical diagnostics on model theta levels (or single levels, e.g. surface).
- **Section 51**: This section duplicates all the variables from Section 34, but on pressure levels.
- **Section 52**: This section duplicates all the variables from Section 50, but on pressure levels.

To use this section 1.ukca_chem_plev must be set to true in the UKCA panel. Additionally, s51i999 must also be output, as this is the Heaviside function that is used to mask missing data from the pressure-level fields (as the UM fieldsfile format has missing data set to zero).

**Solution to Task 3.1**
Surface (20m) UKCA CO (x-y)  

1000hPa UKCA CO (x-y)

You were given the task

- Output 3-hourly-mean UKCA CO (section 34 item 010) field to the UPA output stream, and then add-in the equivalent diagnostic on pressure levels using the DP27CCM domain

and were given the hints

- You will need to remember to run the TidyStashTransform macro once you have added the diagnostics in the STASH panel.

For a working Rose suite that has completed this task, please see

- ARCHER: u-as292859777
- vm: u-as297859775

The specific Rose changes made are:

- ARCHER: https://code.metoffice.gov.uk/trac/roses-u/changeset/59777/a/s/2/9/2/trunk
- vm: https://code.metoffice.gov.uk/trac/roses-u/changeset/59775/a/s/2/9/7/trunk

ARCHER:

```bash
Index: app/um/rose-app.conf
===================================================================
--- app/um/rose-app.conf (revision 59739)  
+++ app/um/rose-app.conf (revision 59777)  
@@ -3287,6 +3287,14 @@
| tim_name='T3HHN'
| use_name='UPA'
+[namelist:umstash_streq(34010_c4b8f1ad)]
+dom_name='DALLTH'
+isec=34
+item=10
+package=
+tim_name='T3HHN'
+use_name='UPA'
+  ![name]umstash_streq(34010_c4b8f1ad]
+dom_name='DALLTH'
+isec=34
+item=10
+package=
+tim_name='T3HHN'
+use_name='UPA'
+  ![name]umstash_streq(51010_5e3d3bef)]
+dom_name='DP27CCM'
+isec=34
+item=10
+package=
+tim_name='T3HHN'
+use_name='UPA'
```

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_3
These differences can be found here:

- **PUMA**: /home/ukca/Tutorial/vn10.9/worked_solutions/Task3.1/Task3.1_rose.patch

If you open the .pa file in Xconv, you should see the following additional fields:

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>
| 3 | 96 | 72 | 27 | 1 | CO MASS MIX RATIO ON PRESS LEVS
| 6 | 96 | 72 | 38 | 1 | CO MASS MIXING RATIO

Sample output from this task can be found at /work/n02/n02/ukca/Tutorial/vn10.9/sample_output/Task3.1/atmosa.pa19810901_00 on ARCHER.

**Checklist**

- [ ] Add new STASH diagnostic requests in: um → namelist → Model Input and Output → STASH Requests and Profiles → STASH Requests.
- [ ] Select the appropriate domain, time, and usage profiles for the new diagnostic(s).
- [ ] Run the TidyStashTransform macro.
- [ ] If outputting UKCA pressure-level diagnostics, remember to make sure that _l_ukca_chem_plev and (possibly) _l_ukca_asad_plev are set to True.

**Tutorial 4**

*Written by Luke Abraham 2017*

UKCA Chemistry and Aerosol vn10.9 Tutorial 4

Contents

1 What you will do in this tutorial
2 Task 4.1: Add new tracers into Rose and UKCA
3 Make a branch
   3.1 Metadata
      3.1.1 Using your new metadata within Rose
4 Rose Changes
   4.1 Include your branch for code changes
   4.2 STASHmaster_A
      4.2.1 Section 51
      4.2.2 Help text
      4.2.3 Use your new STASHmaster file in Rose
   4.3 Initialise your new tracer(s)
   4.4 Output your new tracers
5 Required UKCA Code Changes
   5.1 ukca_nmspec_mod.F90
   5.2 ukca_chem_master.F90
   5.3 ukca_set1defs.F90
   5.4 ukca_constants.F90
   5.5 ukca_cspecies.F90
6 Run your suite
7 Solution to Task 4.1
8 Checklist

What you will do in this tutorial

In this tutorial you will learn how to make the required changes in Rose and in UKCA to include new chemical tracers. As you learn how to do this you will add two new tracers to the UKCA tutorial suite you have already been running.

Task 4.1: Add new tracers into Rose and UKCA

Task 4.1: Add in two new tracers in to slots 64 and 65 for the StratTrop chemistry scheme. The tracer in slot 64 will be called **ALICE** and the tracer in slot 65 will be called **BOB** and initialise these tracers to **1.000000e-12**. You should also output these two tracers through the **UPA** stream in STASH as 3-hour means. In UKCA, you should set the conversion factor for each of these to 1.0.

Make a branch

As UKCA is a framework, it must have the infrastructure around it to allow it to work, with the infrastructure being the UM itself. In this tutorial you will add in two new chemical tracers. In past this was done in two steps, but in Rose this must be done in a single step from vn10.6 onwards. Even though these tutorials are at vn10.4, we will follow the same steps as for vn10.6.

For this tutorial you will need to make a new branch in the usual way. You should do this now.

You will have learned how to make UM branches in the NCAS-CMS FCM Tutorial (https://puma.nerc.ac.uk/trac/UM_TUTORIAL). However, now you should use `um.x_tr` as the source of your branch.

For example, first you should make a ticket on the Met Office SRS Trac pages (https://code.metoffice.gov.uk/trac/um/newticket) (login required) and then make the branch like so

```
fcm branch-create --type dev -k ticket_number your_branch_name fcm:um.x_tr@vn10.9
```

before checking-out your branch by

```
fcm checkout fcm:um.x_br/dev/userid/vn10.9_your_branch_name
```

More information on FCM can be found at the NCAS-CMS FCM pages (http://cms.ncas.ac.uk/wiki/Fcm) or the FCM User Guide (http://metomi.github.io/fcm/doc/user_guide/)

Metadata

In Rose, the GUI appearance is controlled by metadata, and it is possible to edit this metadata to add new variables to panels. We will not do this in this tutorial, but you will need to change where Rose looks for metadata, as the STASHmaster_A is also metadata.

Using your new metadata within Rose

To pick-up the metadata changes, you should edit the `meta` path in `um` to point to
um-atmos/HEAD

From now on, when you want to open your Rose suite for editing, instead of doing it through rosie, you should instead open the job for editing by going to your $HOME/roses/[SUITE-ID] directory, and opening it using the command

```
rose edit -M /path/to/vn10.9_your_branch_name/rose-meta
```

If you don't do this, the STASHmaster changes will not be picked up, and you won't be able to select the tracers.

**Rose Changes**

**Include your branch for code changes**

Although we haven't made any code changes yet, you will (at some point) need to include your branch in Rose so that the UKCA code changes can be picked-up and compiled. To do this go to fcm_make → env → Sources and add a new branch by clicking the plus symbol in the um_sources section. You should then put

```
/path/to/vn10.9_your_branch_name
```

if you wish to run from a working copy, and

```
branches/dev/[your MOSRS username]/vn10.9_your_branch_name
```

if you want to run from the repository. You can specify specific revision numbers by putting REV at the end of this line (where REV is the revision number, e.g. 12345 etc.).

**STASHmaster_A**

In Rose-based jobs, adding new UKCA tracers is done through editing the STASHmaster_A file directly. Take a look at the STASHmaster_A file, found in

```
/path/to/vn10.9_your_branch_name/rose-meta/um-atmos/HEAD/etc/stash/STASHmaster/STASHmaster_A
```

This is a large file, as it contains all the possible STASH items. Scroll down the file until you get to section 34, and you will see how the tracers are defined. The key entries to consider are highlighted in red. All entries are fixed-width, so when editing the file, ensure not to change the spacing, otherwise you will get errors.

```
# Section 34 UKCA Chemistry
#===============================================================================
# 1  34  03 MASS MIXING RATIO AFTER TIMESTEP
# 2  0  1  1  2  40  11  0  0  0  0
# 3 000000000000000000000000000001111110 00000000000000000000000000001 1
# 5  0  2101  0  65  0  0  0  0  0  0  0
#===============================================================================
```

The meaning of these entries is:

- **First Line:**
  - 34: The STASH section, in this case Section 34, UKCA prognostics.
  - 1: The STASH item, in this case, the first item in s34, ozone.
  - 03 MASS MIXING RATIO AFTER TIMESTEP: The name of the field in the STASHmaster file. This can be anything descriptive, but it isn't read by the model, and is only considered in Rose (or e.g. Xconv).

- **Second Line:**
  - 2: The Space Code, which tells the UM whether the field should be held in the restart dump or not. For prognostics, this should be 2 (which means that they should be stored in the restart dump), but for diagnostics that are only passed to output files, this should be 0.

- **Third Line:**
  - 00000000000000000000000111110: These are Options Codes which determine which scheme the STASH field (i.e. tracer in this case) are valid for. These are counted from the right, n1 to n30, and for UKCA only the first 8 are used. These have the following meaning:
    - n1: Age-of-air tracer (ageair) or aerosol tracers
    - n2: Standard Tropospheric Chemistry scheme using the Backward Euler solver (trop)
    - n3: Regional Air Quality Chemistry scheme using the Backward Euler solver (rad)
    - n4: Tropospheric Chemistry with Isoprene scheme using the Newton-Raphson solver (tropisop)
    - n5: Stratospheric-Tropospheric Chemistry using the Newton-Raphson solver (strattrop, also known as CheST)
    - n6: Standard Stratospheric Chemistry using the Newton-Raphson solver (strat)
    - n7: Offline-oxidants scheme (to drive GLOMAP-mode) using the Newton-Raphson solver (offline)
    - n8: Offline-oxidants scheme (to drive GLOMAP-mode) using the Newton-Raphson solver (offline_be)

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_4
Therefore, looking at the option code for ozone (s34i001), it is valid for the trop, raq, tropisop, stratrop, and strat chemistry schemes.

**Fifth Line:**

- **2101**: This is the PP Field Code and should be unique to each tracer. Here these begin counting at 2101 (s34i001) and end at 2356 (s34i256).

So to make a STASH entry for a single tracer called ALICE in slot s34i064 that is only valid in the StratTrop scheme, the STASHmaster_A entry would look like

```plaintext
# Section 51 UKCA Chemistry on pressure levels
#===============================================================================
# 1  1  51  1  O3 MASS MIXING RATIO ON PRESS LEVS  |
# 2  0  0  1  1  1  3  1  1  2  0  0  1  1  3  |
# 000000000000000000000000111110 000000000000000000001 | 3 |
# 5  0  2101  0  8  0  0  0  0  0  0  0  0  0  0  0 |
# ================================================================================
```

**Remember:** You need to add 2 tracers, ALICE in s34i064 and BOB in s34i065.

You should make these changes to STASHmaster_A, and then `fcmm` commit these changes. This is important, otherwise ARCHER will not be able to see the changes you have made once you have included your new STASHmaster_A file in Rose.

**Section 51**

All STASH entries in Section 34 need a corresponding pressure-level field to be added to section 51. The equivalent STASHmaster_A entry for s51 ozone is:

```plaintext
# Section 51 UKCA Chemistry on pressure levels
#===============================================================================
# 1  1  51  1  O3 MASS MIXING RATIO ON PRESS LEVS  |
# 2  0  0  1  1  1  3  1  1  2  0  0  1  1  3  |
# 000000000000000000000000111110 000000000000000000000000000001 | 3 |
# 5  0  2101  0  8  0  0  0  0  0  0  0  0  0  0  0 |
# ================================================================================
```

You will notice that there are a number of changes to various codes, and this is to do with defining the grid that the diagnostic is valid on, whether it can be held in the dump, etc. For more information on what these codes mean, please see Appendix C in UMDP C4, which can be obtained from the SRS here (password required) ([https://code.metoffice.gov.uk/doc/um/latest/umdp.html](https://code.metoffice.gov.uk/doc/um/latest/umdp.html)). When making a new entry, the easiest thing to do is copy an existing entry and make the required changes to the item, name, PP field code, and option codes.

**Remember:** You need to add 2 tracers, ALICE in s34i064 and BOB in s34i065.

**Help text**

You should also add appropriate help text into the `vn10.4_your_branch_name/rose-meta/um-atmos/HEAD/etc/stash/STASHmaster/STASHmaster-meta.conf` file, e.g.

```plaintext
{stashmaster:code(34064)}
description=ALICE MASS MIXING RATIO AFTER TIMESTEP
help=UKCA Tutorial Tracer ALICE Mass Mixing Ratio in kg/kg(Air)
```

and

```plaintext
{stashmaster:code(51064)}
description=ALICE MASS MIX RATIO ON PRESS LEVS
help=UKCA Tutorial Tracer ALICE Mass Mixing Ratio in kg/kg(Air)
```

**Remember:** You need to add 2 tracers, ALICE in s34i064 and BOB in s34i065.

**Use your new STASHmaster file in Rose**

Before you make these changes you should first save and close your suite.

To get Rose to recognise your STASHmaster file, you will need to make a number of changes:

1. Point the `um` metadata to `um-atmos/HEAD` and edit Rose using `rose edit -M /path/to/vn10.9_your_branch_name/rose-meta` (you should already be doing this after following the instructions in the metadata section above).
2. You need to open the `$HOME/roses/[SUITE-ID]/app/um/rose-app.conf` file in a text editor (e.g. vim, emacs, nedit etc.) and add the line `STASHMASTER=STASHmaster` inside the `[env]` block.
3. You need to open the `$HOME/roses/[SUITE-ID]/rose-suite.conf` file in a text editor and add the following lines at the top of the file:

```plaintext
[file:app/um/file/STASHmaster]
source=fcmm:um.xm_br/dev/[your MOSRS userid]/vn10.9_your_branch_name/rose-meta/um-atmos/HEAD/etc/stash/STASHmaster
```

The array is called `these are contained in the first 256 slots`. Inside UKCA there is a master mapping array that tells the UKCA code which transported species are placed into a particular item number in section 34. At vn10.9 it looks like this:

```
03   'NO   'N3      'N2O      'N2O5     &
H2O2N2 'HONO2  'H2O2    'CH4     'CO       &!10
HCHO  'MeOOH  'HONO    'C2H6    'EtOOH    &
MeCHO  'PAN    'C3H8    'n-PrOOH  'i-PrOOH  &!20
EtCHO 'Me2CO   'MeCOCH2OH 'PPAN    'MeONO2   &
Q_3_S 'CSH8    'ISOOH   'ISON    'MACR     &!30
MACROOH 'MFPAN   'MACE    'MO1Y    'NADL     &
HCOOH 'MeCO3H   'H2O     'ISO2    'H2O      &!40
Cl   'C1O    'C12O2   'OC1O    'Br       &
BrO  'BrCl   'BrONO2  'N2O     'HCl      &!50
HOC1 'HBr    'HOBr    'C1ON2    'CFC13    &
CF2Cl2 'MeBr    'N       'O(3P)   'MACRO2   &!60
MeCl  'CF2ClBr 'CC14    'CF2ClCFC12 'CF2Cl2   &
MeCC13 'CF3Br   'H2OS    'CH2Br2    'H2       &!70
DNS  'SO2     'H2SO4   'MSA      'DMSO     &
NH3  'CS2     'CO5     'H2S     'H        &!80
OH  'HO2     'MeO0    'EtO0    'MeCO3    &
n-PrOO 'i-PrOO  'EtCO3   'MeCOCH200  'MeOH    &!90
Monoterp 'Sec_Org 'SEESQUIPER 'SO3     'AROM     &
O(3P)_S 'O1(1D)_S 'NO2     'BrO      'HCl      &!100
Nuc_SOL_ND 'Nuc_SOL_SU 'Ait_SOL_ND 'Ait_SOL_SU 'Ait_SOL_BC &
Ait_SOL_OC 'Acc_SOL_ND 'Acc_SOL_SU 'Acc_SOL_BC 'Acc_SOL_OC &!110
Acc_SOL_SS 'Acc_SOL_DU 'Cor_SOL_ND 'Cor_SOL_SU 'Cor_SOL_BC &
Cor_SOL_OC 'Cor_SOL_SS 'Cor_SOL_DU 'Cor_INS_ND 'Ait_INS_BC &!120
Ait_INS_OC 'Acc_INS_ND 'Acc_INS_DU 'Cor_INS_ND 'Cor_INS_DU &
Nuc_SOL_OC 'Ait_SOL_SS 'Nuc_SOL_SO 'Ait_SOL_SO 'Acc_SOL_SO &!130
Cor_SOL_SO 'Nuc_SOL_NH 'Ait_SOL_NH 'Acc_SOL_NH 'Cor_SOL_NH &
Nuc_SOL_NT 'Ait_SOL_NT 'Acc_SOL_NT 'Cor_SOL_NT 'XXX &!140
Anthr_Prec 'Bio_Prec 'Anth_Cond 'Bio_Cond 'XXX &
XXX   'XXX   'XXX   'PASSIVE O3   'AGE OF AIR &!150
RETIRED  'RETIRED RETIRED 'RETIRED RETIRED &
RETIRED  'RETIRED RETIRED 'RETIRED RETIRED &!160
RETIRED  'RETIRED RETIRED 'RETIRED RETIRED &
RETIRED  'RETIRED RETIRED 'RETIRED RETIRED &!170
RETIRED  'RETIRED XXX   'XXX   'XXX &
XXX   'XXX   'XXX   'XXX   'XXX &!180
XXX   'XXX   'XXX   'XXX   'XXX &
XXX   'XXX   'XXX   'XXX   'XXX &!190
XXX   'XXX   'XXX   'XXX   'XXX &
XXX   'XXX   'XXX   'XXX   'XXX &!200
XXX   'XXX   'XXX   'XXX   'XXX &
XXX   'XXX   'XXX   'XXX   'XXX &!210
XXX   'XXX   'XXX   'XXX   'XXX &
XXX   'XXX   'XXX   'XXX   'XXX &
XXX   'XXX   'XXX   'XXX   'XXX &
```

This is called `nm_spec` and it can be found in `ukca_nmspec_mod.F90`. At vn10.9, it looks like this:

```
!13
```

You need to add 2 tracers, ALICE in s34i064 and BOB in s34i065.

**Required UKCA Code Changes**

Inside UKCA there is a master mapping array that tells the UKCA code which transported species are placed into a particular item number in section 34. At vn10.9 these are contained in the first 256 slots.

This is called `nm_spec` and it can be found in `ukca_nmspec_mod.F90`. At vn10.9, it looks like this:

```
!13
```

To output your tracers, go to the STASH panel and output your tracers as described in the What is STASH? tutorial.

**Remember:** You need to add 2 tracers, ALICE in s34i064 and BOB in s34i065.

You need to do this for each of the tracers you are adding (you can also clone-existing entries). When you have finished, you will need to go to the STASH Requests panel (see the STASH tutorial) and run the TidyStashTransform macro to correctly generate the required index value for each tracer initialisation.

**Output your new tracers**

To add new fields, right click anywhere within the table and click Add new section. This will make a new entry with index 1, if you add another entry, it will be given index 2 etc. Right-click on this and click on View namelist:items(X) (where X is the index number). This will open a new tab where you can select the STASH section and item for the field you want to initialise using the drop-down menus. Some UKCA tracers need to be given full 3D initial conditions in ancillary file format. However, for this example we will initialise the tracer to a small number, \(1.000000 \times 10^{-12}\). To do this, select the radio button for **Set to a specified constant value** and put \(1.000000e-12\) in the dialog box labelled `user_prog_rconst` at the bottom of the panel.

**Initiate your new tracer(s)**

Your new tracer(s) is a prognostic variable, as therefore the UM requires prior knowledge of the values it has to be able to run properly. To do this you need to go to um >>> namelist >>> Reconfiguration and Ancillary Control >>> Configure ancils and initialise dump fields. Here you will see a list of fields. You can change the order of these by e.g. STASH number, as this often makes it easier to view.

To add new fields, right click anywhere within the table and click Add new section. This will make a new entry with index 1, if you add another entry, it will be given index 2 etc. Right-click on this and click on View namelist:items(X) (where X is the index number). This will open a new tab where you can select the STASH section and item for the field you want to initialise using the drop-down menus. Some UKCA tracers need to be given full 3D initial conditions in ancillary file format. However, for this example we will initialise the tracer to a small number, \(1.000000 \times 10^{-12}\). To do this, select the radio button for **Set to a specified constant value** and put \(1.000000e-12\) in the dialog box labelled `user_prog_rconst` at the bottom of the panel.

With this example we will initialise the tracer to a small number, \(1.000000 \times 10^{-12}\). To do this, select the radio button for **Set to a specified constant value** and put \(1.000000e-12\) in the dialog box labelled `user_prog_rconst` at the bottom of the panel.

The array is called `nm_spec` and it can be found in `ukca_nmspec_mod.F90`. At vn10.9, it looks like this:

```
!13
```

To output your tracers, go to the STASH panel and output your tracers as described in the What is STASH? tutorial.

**Remember:** You need to add 2 tracers, ALICE in s34i064 and BOB in s34i065.

**Required UKCA Code Changes**

Inside UKCA there is a master mapping array that tells the UKCA code which transported species are placed into a particular item number in section 34. At vn10.9 these are contained in the first 256 slots.

This is called `nm_spec` and it can be found in `ukca_nmspec_mod.F90`. At vn10.9, it looks like this:
This array is case sensitive and each string is a fixed length of 10 characters, with the whitespace being made up of spaces and not another type of white space (e.g. tabs).

When adding new tracers you should not overwrite tracers that are already in use by the chemistry scheme that you are using (e.g. StratTrop etc.), and it is best to also take care and avoid tracers in use by other schemes, if possible. You can see the tracers used by each scheme by looking in the chch_defs array at the top of the ukca_chem_master.F90 module.

Here you should put a new species, ‘ALICE’, into tracer slot 64 (current specified as ‘CF2ClCFCl2’, which is not in use by any scheme at the moment).

Remember: You need to add 2 tracers, ALICE in s34i064 and BOB in s34i065.

Therefore to add-in the ALICE species, we should insert a line similar to this one:

chch_t1(116,’ALICE’,1,’TR’,’,’,0,0,0,ST,0,0,107), &

into chch_defs_master, remembering to also increment the size of this array, given by n_chch_master.

Remember: You need to add 2 tracers, ALICE in s34i064 and BOB in s34i065.
The IF block tests against L_ukca_strattrop and L_ukca_achem (which determines whether or not you require the additional chemistry used to drive the GLOMAP-mode aerosol scheme). You should increment the value of n_chem_tracers by the number of chemical tracers that you are adding in both sections of the IF block, as this tracer is not defined to be for aerosol chemistry only. If your additional tracers are aerosol chemistry additions then you should increment the value of n_aero_tracers.

If you are adding to a different chemistry scheme then you will need to make these changes accordingly.

Remember: You need to add 2 tracers, ALICE in s34i064 and BOB in s34i065.

ukca_constants.F90

The unit of the tracers is kg(species)/kg(air) (i.e. mass mixing ratio, or mmr), but inside UKCA these species are converted to volume mixing ratio (or vmr). To enable UKCA to do this you will need to add the conversion factor(s) for your new tracer(s) into the code. This is done in the ukca_constants.F90 module.

Open this file and add the required conversion factor(s). The naming convention for these is $M_{\text{species}}$ is the molecular mass of the new species in g/mol, and $C_{\text{species}}$ for the conversion factor from vmr to mmr (calculated as $M_{\text{species}}/M_{\text{air}}$, where $M_{\text{air}}=28.97$). In actuality, only the $C_{\text{species}}$ listing is always required, although you may need to add the $M_{\text{species}}$ value later if you are emitting into the new species that you are adding.

For example:

```
REAL, PARAMETER :: C_ALICE = 1.0000
```

Remember: You need to add 2 tracers, ALICE in s34i064 and BOB in s34i065.

ukca_cspecies.F90

After you have added in the $C_{\text{species}}$ conversion factors, you will need to tell UKCA to use them for your species. To do this you need to edit the ukca_cspecies.F90 module, which contains code which constructs the $c_{\text{species}}$ array of conversion factors for the advected tracers. This contains a subroutine called ukca_calc_cspecies which has a long block of code that you need to edit to add an entry like this:

```
WHERE (advt == 'ALICE     ') c_species = C_ALICE
```

The advt array is automatically generated by UKCA at run-time from the chch_defa_scheme chemistry scheme definition you edited earlier, so your new tracer(s) will exist within it. You need to add in a new line for each of your tracers which sets the value of the $c_{\text{species}}$ array to your individual $C_{\text{species}}$ parameter. Add the line(s) in at the end of the block.

Remember: You need to add 2 tracers, ALICE in s34i064 and BOB in s34i065.

Run your suite

Now that you have made the required changes to Rose and in your branch, please run your suite. You may find that it fails on the, fcm_make2, recon, or atmos jobs. To find the errors and see output, you should either right-click the failed task in Gcylc and view the output, or go to the

```
/home/n02/n02/[your ARCHER userid]/cylc-run/[SUITE-ID]
```

directory on ARCHER. Within this there are a number of directories. To see the log files, cd into

```
log/job/1/[JOB NAME]/NN
```

and view the job.err or job.out files. When the suite successfully runs, the fieldsfile output can usually be found in

```
/work/n02/n02/[your ARCHER userid]/cylc-run/[SUITE-ID]/work/1/atmos/
```

Numerical Noise
Numerical noise in the ALICE tracer (s34i064)

Numerical noise in the BOB tracer (s34i065)

When your suite has run successfully, and you view the fields for your new tracer(s) in Xconv, you may find that they have a speckled appearance, rather than being constant. This is due to numerical noise that has been introduced as the tracer is run through the UKCA chemical solver. Once the field is converted to 32-bit this should disappear. It will also cease to become apparent when emissions and reactions are applied to these tracers.

**Solution to Task 4.1**

You were given the task

- Add in two new tracers in to slots 64 and 65 for the StratTrop chemistry scheme. The tracer in slot 64 will be called **ALICE** and the tracer in slot 65 will be called **BOB** and initialise these tracers to **1.000000e−12**. You should also output these two tracers through the **UPA** stream in STASH as 3-hour means.

In UKCA, you should set the conversion factor for each of these to 1.0.

For a working Rose suite that has completed this task, please see

- **ARCHER:** u-as292859818
- **vm:** u-as297859815

The specific Rose changes made are:

- **ARCHER:** [https://code.metoffice.gov.uk/trac/roses-u/changeset/59818/a/s/2/9/2/trunk](https://code.metoffice.gov.uk/trac/roses-u/changeset/59818/a/s/2/9/2/trunk)
- **vm:** [https://code.metoffice.gov.uk/trac/roses-u/changeset/59815/a/s/2/9/7/trunk](https://code.metoffice.gov.uk/trac/roses-u/changeset/59815/a/s/2/9/7/trunk)

**ARCHER:**

```plaintext
Index: app/um/rose-app.conf
===================================================================
--- app/um/rose-app.conf   (revision 59777)
+++ app/um/rose-app.conf   (revision 59818)
@@ -1,4 +1,4 @@
-meta=um-atmos/vn10.9
+meta=um-atmos/HEAD

[command]
default=um-atmos
@@ -24,6 +24,7 @@
RECON_KEEP_MPP_STDOUT=true
RUNID=atmos
SPECTRAL_FILE_DIR=$UM_INSTALL_DIR/vn$VN/ctldata/spectral/ga7
+STASHMASTER=STASHmaster

[file:$DATAM]
mode=mkdir
@@ -644,6 +645,18 @@
+!user_prog_ancil_stash_req=0
+!user_prog_r const=0
+![namelist:items(3e9a6939)]
+ancilfilename=
+domain=1
+!!interval=0
+!!netcdf_varname=
+!!period=1
```

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_4
vm:

Index: app/fcm_make/rose-app.conf
===================================================================
--- app/fcm_make/rose-app.conf (revision 59777)
+++ app/fcm_make/rose-app.conf (revision 59818)
@@ -42,4 +42,4 @@
 stash_version=1A
timer_version=3A
 um_rev=vn10.9
-um_sources=
+um_sources=branches/dev/lukeabraham/vn10.9_UKCA_Tutorial_Solns@46535

Index: rose-suite.conf
===================================================================
--- rose-suite.conf (revision 59777)
+++ rose-suite.conf (revision 59818)
@@ -1,3 +1,6 @@
 [file:app/um/file/STASHmaster]
+source=fcm:um.xm_br/dev/lukeabraham/vn10.9_UKCA_Tutorial_Solns/rose-meta/um-atmos/HEAD/etc/stash/STASHmaster@HEA

[jinja2:suite.rc]
ARCHER_GROUP='n02-chem'
ARCHER_QUEUE='short'

Index: app/um/rose-app.conf
===================================================================
--- app/um/rose-app.conf (revision 59775)
+++ app/um/rose-app.conf (revision 59815)

--- app/um/rose-app.conf (revision 59775)
+++ app/um/rose-app.conf (revision 59815)

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_4
```bash
@ -1,4 +1,4 @@
-meta=um-atmos/vn10.9
+meta=um-atmos/HEAD

[command]
  default=um-atmos
@@ -26,6 +26,7 @@
  RECON_STDOUT_FILE=pe_output/atmos.fort6.pe
  RUNID=atmos
  SPECTRAL_FILE_DIR=$UMDIR/vn$VN/ctldata/spectral/ga7
+STASHMASTER=STASHmaster
  UM_THREAD_LEVEL=MULTIPLE

[file:$DATAM]
@@ -647,6 +648,18 @@
  !!user_prog_ancil_stash_req=0
  !!user_prog_rconst=0
+[
+  namelist:items(3e9a6939)]
+  ancilfilename=
+  +domain=1
+  +!!interval=0
+  +!!netcdf_varname=
+  +!!period=1
+  +source=6
+  +stash_req=34064
+  +update_anc=.false.
+  +!!user_prog_ancil_stash_req=0
+  +user_prog_rconst=1.000000e-12
+[
+  namelist:items(4a4f86c3)]
  ancilfilename='$UMDIR/ancil/atmos/n48e/orca1/seaice/reynolds/1981_2012_360/v1/qrclim.seaice'
  +domain=1
@@ -731,6 +744,18 @@
  +!!interval=0
  +!!netcdf_varname=
  +!!period=1
  +source=6
  +stash_req=34065
  +update_anc=.false.
  +!!user_prog_ancil_stash_req=0
  +user_prog_rconst=1.000000e-12
+[
+  namelist:items(c3e3b7ad)]
+  +ancilfilename=
+  +domain=1
+  +!!interval=0
+  +!!netcdf_varname=
+  +!!period=1
+  +source=6
+  +stash_req=34065
+  +update_anc=.false.
+  +!!user_prog_ancil_stash_req=0
+  +user_prog_rconst=1.000000e-12
+[
+  namelist:items(c54d9b29)]
  ancilfilename='$UMDIR/ancil/atmos/n48e/orca1/land_sea_mask/etop01/v1/qrparm.mask'
  +domain=1
@@ -3887,6 +3912,22 @@
  +tim_name='T3HMN'
  +use_name='UPA'
+[
+  namelist:umstash_streq(34064_487e535d)]
+  +dom_name='DALLTH'
+  +isec=34
+  +item=64
+  +package=
+  +tim_name='T3HMN'
+  +use_name='UPA'
+[
+  namelist:umstash_streq(34065_6dc07351)]
+  +dom_name='DALLTH'
+  +isec=34
+  +item=65
+  +package=
+  +tim_name='T3HMN'
+  +use_name='UPA'
+[
+  namelist:umstash_streq(34106_ebb7bb67)]
  +dom_name='DALLTH'
  +isec=34
Index: app/fcm_make/rose-app.conf
===================================================================
--- app/fcm_make/rose-app.conf (revision 59775)
+++ app/fcm_make/rose-app.conf (revision 59815)
@@ -42,4 +42,4 @@
```

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_4

These ARCHER differences can be found in the file `/home/ukca/Tutorial/vn10.9/worked_solutions/Task4.1/Task4.1_rose.patch` on PUMA.

The specific UM changes made are:

```fortran
Index: src/atmosphere/UKCA/ukca_setd1defs.F90
===================================================================
--- src/atmosphere/UKCA/ukca_setd1defs.F90 (revision 46527)
+++ src/atmosphere/UKCA/ukca_setd1defs.F90 (revision 46535)
@@ -321,7 +321,7 @@
     ('NO      ','CH4       ','CO       ','HCHO     ', &
     'C2H6     ','C3H8     ','Me2CO    ','MeCHO    ', &
     'C5H8     ','NO_aircrft'/)
-    n_chem_tracers = 71 ! No chem tracers
+    n_chem_tracers = 73 ! No chem tracers
    nr_therm     = 220 ! thermal reactions
    nr_phot      = 55  ! photolytic (ATA)
@@ -337,7 +337,7 @@
     'SO2_high  ','NH3       ','DMS       ','SO2_nat   ', &
     'BC_biomass','OC_biomass','NO_aircrft'/)
-    n_aero_tracers = 12 ! No aerosol tracers
+    n_aero_tracers = 12 ! No aerosol tracers
    IF (L_ukca_trophet) THEN
    nr_therm     = 241 ! thermal reactions
-ELSE
+ELSE
```

```fortran
Index: src/atmosphere/UKCA/ukca_chem_master.F90
===================================================================
--- src/atmosphere/UKCA/ukca_chem_master.F90 (revision 46527)
+++ src/atmosphere/UKCA/ukca_chem_master.F90 (revision 46535)
@@ -79,7 +79,7 @@
     ! define below the chemistry schemes
     !-INTEGER, PARAMETER :: n_chch_master = 148 ! number of known species
 +INTEGER, PARAMETER :: n_chch_master = 150 ! number of known species
     INTEGER, PARAMETER :: n_het_master  =  10 ! number of heterogeneous reactions
     INTEGER, PARAMETER :: n_dry_master  =  57 ! number of dry deposition reactions
     INTEGER, PARAMETER :: n_wet_master  =  49 ! number of wet deposition reactions
@@ -376,7 +376,9 @@
     chch_t1(112,'TOLP1     ',1,'SS        ','          ',0,0,0,R,0,0,107),&
     chch_t1(113,'MEMALD    ',1,'TR        ','          ',0,0,0,R,0,0,107),&
-    chch_t1(115,'oXYLENE   ',1,'TR        ','          ',0,0,1,R,0,0,107)/)
+    chch_t1(115,'oXYLENE   ',1,'TR        ','          ',0,0,1,R,0,0,107),&
+    chch_t1(116,'ALICE     ',1,'TR        ','          ',0,0,0,ST,0,0,107),&
+    chch_t1(117,'BOB       ',1,'TR        ','          ',0,0,0,ST,0,0,107)/)
! Heterogeneous chemistry
! Columns take the following meanings:
Index: src/atmosphere/UKCA/ukca_constants.F90
===================================================================
--- src/atmosphere/UKCA/ukca_constants.F90 (revision 46527)
+++ src/atmosphere/UKCA/ukca_constants.F90 (revision 46535)
@@ -272,6 +272,10 @@
     REAL, PARAMETER :: c_ALICE      = 1.0000
     REAL, PARAMETER :: c_BOB        = 1.0000
     REAL, PARAMETER :: c_isosvoc2   = 2.3473 ! as C5H8
+    REAL, PARAMETER :: c_isosoa     = 4.4874 ! 130.0
+    REAL, PARAMETER :: c_ALICE      = 1.0000
+    REAL, PARAMETER :: c_BOB        = 1.0000
```

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_4
molecular masses in g/mol of emitted species,
for budget calculations
REAL, PARAMETER :: m_ho2 = 33.007

Index: src/atmosphere/UKCA/ukca_nmspec_mod.F90

--- src/atmosphere/UKCA/ukca_nmspec_mod.F90 (revision 46527)
+++ src/atmosphere/UKCA/ukca_nmspec_mod.F90 (revision 46535)
@@ -86,7 +86,7 @@
     'BrO' 'BrCl' 'BrONO2' 'H2O' 'HCl' , & 150
     'HOCI' 'HBr' 'HOBr' 'ClONO2' 'CFCl3' , &
     'CF2Cl2' 'MeBr' 'N' 'O(3P)' 'MACRO2' , & 160
-    'MeC1' 'CF2ClBr' 'CC14' 'CF2ClCFC12' 'CF2FC1' , &
+    'MeC1' 'CF2ClBr' 'CC14' 'ALICE' 'BOB' , &
     'MeCCl3' 'CF3Br' 'H2O' 'CH2Br2' 'H2' , & 170
     'DMS' 'SO2' 'H2SO4' 'MSA' 'DMSO' , &
     'NH3' 'CS2' 'COS' 'H2S' 'H' , & 180

Index: src/atmosphere/UKCA/ukca_cspecies.F90

--- src/atmosphere/UKCA/ukca_cspecies.F90 (revision 46527)
+++ src/atmosphere/UKCA/ukca_cspecies.F90 (revision 46535)
@@ -296,6 +296,9 @@
     WHERE (advt == 'ORGNIT ') c_species = c_orgnit
     WHERE (advt == 'PASSIVE O3') c_species = 1.0
     WHERE (advt == 'AGE OF AIR') c_species = 1.0
+! UKCA Tutorial Tracers:
+    WHERE (advt == 'ALICE     ') c_species = c_ALICE
+    WHERE (advt == 'BOB       ') c_species = c_BOB
+!
!
+!
+!
+!
+!
!

Index: rose-meta/um-atmos/HEAD/etc/stash/STASHmaster/STASHmaster-meta.conf

--- rose-meta/um-atmos/HEAD/etc/stash/STASHmaster/STASHmaster-meta.conf (revision 46527)
+++ rose-meta/um-atmos/HEAD/etc/stash/STASHmaster/STASHmaster-meta.conf (revision 46535)
@@ -16215,6 +16215,14 @@
     [stashmaster:code(34065)]
     +description=ALICE MASS MIXING RATIO AFTER TIMESTEP
     +help=UKCA Tutorial Tracer ALICE Mass Mixing Ratio in kg/kg(Air)
+    =on pressure levels
     [stashmaster:code(51064)]
     +description=H2 MASS MIXING RATIO AFTER TIMESTEP
     +help=Hydrogen Mass Mixing Ratio in kg/kg(Air)
+    =on pressure levels
     [stashmaster:code(51065)]
     +description=BOB MASS MIXING RATIO AFTER TIMESTEP
     +help=UKCA Tutorial Tracer BOB Mass Mixing Ratio in kg/kg(Air)
+    =on pressure levels
     [stashmaster:code(51070)]
     +description=ALICE MASS MIXING RATIO AFTER TIMESTEP
     +help=UKCA Tutorial Tracer ALICE Mass Mixing Ratio in kg/kg(Air)
+    =on pressure levels
     [stashmaster:code(51075)]
     +description=BOB MASS MIXING RATIO AFTER TIMESTEP
     +help=UKCA Tutorial Tracer BOB Mass Mixing Ratio in kg/kg(Air)
+    =on pressure levels

Index: rose-meta/um-atmos/HEAD/etc/stash/STASHmaster/STASHmaster_A

--- rose-meta/um-atmos/HEAD/etc/stash/STASHmaster/STASHmaster_A (revision 46527)
+++ rose-meta/um-atmos/HEAD/etc/stash/STASHmaster/STASHmaster_A (revision 46535)
@@ -15942,6 +15942,21 @@
     +description=H2 MASS MIXING RATIO AFTER TIMESTEP
     +help=Hydrogen Mass Mixing Ratio in kg/kg(Air)
+    =on pressure levels
     +
     [stashmaster:code(51065)]
     +description=BOB MASS MIXING RATIO AFTER TIMESTEP
     +help=UKCA Tutorial Tracer BOB Mass Mixing Ratio in kg/kg(Air)
+    =on pressure levels
     +
     [stashmaster:code(51070)]
     +description=ALICE MASS MIXING RATIO AFTER TIMESTEP
     +help=UKCA Tutorial Tracer ALICE Mass Mixing Ratio in kg/kg(Air)
+    =on pressure levels
     +

Index: rose-meta/um-atmos/HEAD/etc/stash/STASHmaster/STASHmaster_A

--- rose-meta/um-atmos/HEAD/etc/stash/STASHmaster/STASHmaster_A (revision 46527)
+++ rose-meta/um-atmos/HEAD/etc/stash/STASHmaster/STASHmaster_A (revision 46535)
@@ -15942,6 +15942,21 @@
     +description=H2 MASS MIXING RATIO AFTER TIMESTEP
     +help=Hydrogen Mass Mixing Ratio in kg/kg(Air)
+    =on pressure levels
     +
     [stashmaster:code(51065)]
     +description=BOB MASS MIXING RATIO AFTER TIMESTEP
     +help=UKCA Tutorial Tracer BOB Mass Mixing Ratio in kg/kg(Air)
+    =on pressure levels
     +
     [stashmaster:code(51070)]
     +description=ALICE MASS MIXING RATIO AFTER TIMESTEP
     +help=UKCA Tutorial Tracer ALICE Mass Mixing Ratio in kg/kg(Air)
+    =on pressure levels
     +

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_4
These differences can be found in the file /home/ukca/Tutorial/vn10.9/worked_solutions/Task4.1/Task4.1_code.patch on PUMA.

If you open the .pa file in Xconv, you should see the following fields:

```plaintext
7    : 96   72   38    1     field2164: Stash code = 34064
8    : 96   72   38    1     field2165: Stash code = 34065
```

Sample output from this task can be found at /work/n02/n02/ukca/Tutorial/vn10.9/sample_output/Task4.1/atmosa.pa19810901_00 on ARCHER.

**Checklist**

- If required, copy an appropriate Rose suite to use to test your changes.
- Make a new ticket on the Met Office SRS Trac pages.
- Make a branch at the required MetUM version using fcm branch-create.
- Checkout your branch using fcm checkout.
- Choose slots for your new tracer(s) from the list in ukca_set_nmspec.F90, and add the correct names for them in these slots.
- Add the tracer(s) to the chch_defs_strattrop array in ukca_chem_master.F90.
- Increase the value of n_chem_tracers in ukca_setd1defs.F90.
- Add a C_species value for each tracer in ukca_constants.F90.
- Append the CASE statement in ukca_cspecies.F90 to add this value to the c_species array.
- fcm commit the changes to your branch.
- Make the required changes to branch's STASHmaster_A file, to add the new tracers.
- Add help text for your tracers in your branch's STASHmaster-meta.conf file.
- Using a text editor, open the app/um/rose-app.conf file from your roses/[SUITE-ID] directory, and add the line STASHMASTER=STASHmaster in the [env] block, then save and close the file.
- Using a text editor, open the rose-suite.conf file from your roses/[SUITE-ID] directory, and add the following lines to the top of the file, before saving and closing it:

```
[[file:app/um/file/STASHmaster]
source=fcm:um.xm_br/dev/[your MOSRS userid]/vnX.Y_your_branch_name/rose-meta/um-atmos/HEAD/etc/stash/STASHmaster]
```

- Edit your suite using rose edit -M /path/to/your/branch/working/copy.
☐ Point the metadata in your suite to um-atmos/HEAD.
☐ Include your branch in your suite at: `fcm_make` ➔ `env` ➔ `Sources`.
☐ Initialise your tracers in your suite at: `um` ➔ `namelist` ➔ `Reconfiguration and Ancillary Control` ➔ `Configure ancils and initialise dump fields`.
☐ Output your tracers in STASH at: `um` ➔ `namelist` ➔ `Model Input and Output` ➔ `STASH Requests and Profiles` ➔ `STASH Requests`.
☐ Run the TidyStashTransform transform macro.
☐ Save your suite.
☐ In the `roses/[SUITE-ID]` directory, run `fcm commit` to commit your changes to the repository.
☐ Run your suite.

Tutorial 5

Written by Luke Abraham 2017


- This page was last modified on 26 April 2018, at 10:18.
UKCA Chemistry and Aerosol vn10.9 Tutorial 5

From UKCA

UKCA Chemistry and Aerosol Tutorials at vn10.9

Contents

1 What you will learn in this Tutorial
2 Task 5.1: Create a new emissions file and use it in your job
3 The netCDF emissions system
   3.1 Metadata
4 Using python to regrid your emissions
   4.1 ARCHER
   4.2 Virtual Machine
   4.3 Using python
   4.4 Example python script
5 Solution to Task5.1
6 Task 5.2: make the required code changes to add your emission into UKCA
7 Rose changes
   7.1 ARCHER
   7.2 Virtual Machine
   7.3 Make your Rose changes
8 Code changes
   8.1 ukca_chem_master.F90
   8.2 ukca_setd1defs.F90
   8.3 ukca_constants.F90
   8.4 get_molmass_mod.F90
   8.5 Emissions diagnostics
      8.5.1 STASHmaster_A and STASHmaster-meta.conf
      8.5.2 get_emdiag_stash_mod.F90
      8.5.3 ukca_emdiags_struct_mod.F90
      8.5.4 ukca_update_emdiagsstruct_mod.F90
      8.5.5 ukca_emiss_diags_mod.F90
      8.5.6 ukca_emiss_ctl_mod.F90
9 Solution to Task 5.2
10 Task 5.3: Output the diagnostic of your new emission
11 Solution to Task 5.3
12 Checklist

What you will learn in this Tutorial

During this tutorial you will learn how to make new UKCA netCDF emissions files. Then you will learn how to add new emissions into UKCA so that they emit into one of your new tracers.

At the end of the previous tutorial you will now know how to create new tracers for use by UKCA. However, after completing the tasks, your tracers will still be empty, as nothing has been put into them. This tutorial will teach you how to create an emissions file that the UKCA will read, and that you can then tell UKCA to use and emit into your tracer(s).

This tutorial will go through the steps needed to make an emission into a tracer which UKCA does not currently emit into. The steps in making the netCDF file will be the same for a species which is currently emitted into, although in this simpler case you would not need to make any code changes.

This example only deals with a single field. For a more detailed examples on changing emissions for all species, please see the Emissions for ACSIS webpage.

Task 5.1: Create a new emissions file and use it in your job

Task 5.1: In the

```
/work/n02/n02/ukca/Tutorial/vn10.9/Task5.1
```

directory on ARCHER there is the file Emissions_of_ALICE.nc which is a 0.5x0.5 degree resolution surface emission field. You should regrid this file to the N48 ENDGame grid, and output it as a netCDF file that has the required netCDF metadata for a climatological surface emission without any diurnal cycle.

The netCDF emissions system

UKCA uses netCDF files to specify the emissions into UKCA species. Using netCDF files means that the metadata in the file can be used to specify various options, such as which diurnal cycle to use.

Metadata

In Global Atmosphere 7.0, UKCA emissions are prescribed using netCDF files. The reason for this move away from the UM ancillary file format

Each emission field in the NetCDF files needs to include the following variable metadata attributes (there is no requirement for the netCDF variable names themselves except that they be no longer than 80 characters):

- **standard_name**: It should be included if an appropriate name is present in the CF Standard Name Table (see http://cfconventions.org/standard-names.html). An example of a valid ‘standard name’ is:

  "tendency_of_atmosphere_mass_content_of_nitrogen_monoxide_due_to_emission"

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_5
you will see that Iris is available:

```
iris                      1.10.0               np19py27_2
```

However, under some circumstances emissions fields need to be reported as kg of a given species (e.g. nitrogen, carbon, sulfur). When that is the case this needs to be indicated in the standard name attribute if possible (see example below), otherwise in the long name attribute. As an example, in the case of an emission field reported as kg of carbon, the attribute units will be "kg m-2 s-1" while the attribute standard name should contain the substring "expressed as carbon" if that is accepted by CF conventions for standard names (see http://cf-pcmdi.llnl.gov/documents/cf-standard-names); otherwise the attribute long name should contain the substring "expressed as carbon". The UM subroutines BASE_EMISS_FACTORS and GET_BASE_SCALING will look for such substrings and apply some conversions if needed. It is therefore essential that the substrings in standard_name and long_name are separated by underscores and white spaces, respectively, as indicated above.

- **hourly_scaling**: (optional) A character attribute used to apply a diurnal cycle to emissions data with daily or lower frequency. Allowed values:
  - none (or attribute not present): no scaling
  - traffic_uk: used for UK air quality studies
  - TNO_MACC_EU_SNAPnn (where nn=01 to 11): Hourly factors of emissions for Europe. Calculated by TNO for the MACC project.
  - diurnal_isopems (for isoprene emissions): use the routine UKCA DIURNAL ISOP EMS to calculate a diurnal cycle using solar zenith angle and latitude to compute the expected number of sunshine hours.

- **daily_scaling**: (optional): A character attribute used to apply a weekly cycle to emissions weekly or lower frequency. Only allowed when the model is using the Gregorian calendar. Allowed values:
  - none (or attribute not present): no scaling
  - traffic_uk: used for UK air quality studies
  - TNO_MACC_GB_SNAPnn (where nn=01 to 11): Hourly factors of emissions for Great Britain. Calculated by TNO for the MACC project.

- **vertical_scaling**: (optional): A character attribute defining the vertical distribution of the emission. This allows the user to supply a 2D field to be applied over multiple levels. Allowed values:
  - surface (or attribute not present): treat as surface emission
  - all_levels, 3D: field is three-dimensional and is provided on model levels (must have the same number of levels as the model)
  - high_level: spread a 2D field over multiple model levels, weighting by model layer thickness to achieve a uniform distribution in height. Must be accompanied by variable attributes of emissions levels at highest level to indicate the model levels over which to distribute the emission.
  - step1: used for air quality simulations, spreads emission over lowest 3 layers of the model; only allowed with 36 model levels.
  - EMEP_modified_SNAPnn (where n=01 to 11): Average vertical profiles for SNAP sectors, similar to implementation for EMEP model. See routine vertical_emiss_factors for details.
  - Bieser_modified_SNAPnn (where n=01 to 11): Average vertical profiles for SNAP sectors, similar to implementation of Bieser et al. (2011) with the SMOKE model and including fugitive emissions. See routine vertical_emiss_factors for details. In addition, each NetCDF file has to include two global attributes with information that is valid for all emission fields present in the file:
    - **update_freq_in_hours**: Integer indicating the frequency (in hours) at which all emission fields present in that file should be read to update emissions().%!values (:;:) in the UKCA code. Update points are calculated relative the model's ancil ref time.
    - **update_type**: Integer number indicating the times at which the data is provided. The same conventions as for ancillary files have been adopted:
      0: Single time
      1: Time series
      2: Periodic time series

When emissions are treated as time series (update_type=1) the user is responsible for creating emission fields which cover the whole period for which the model is run; otherwise the model will not find the time registers to do interpolations and will stop with error. When emissions are periodic (update_type=2) each emission field has to include exactly 12 monthly average emissions (Jan, Feb, ..., Dec). Other attributes, in particular some global attributes, as well as some additional fields (e.g. a variable indicating the type of grid mapping) should be present in the files to comply with CF conventions for NetCDF data, but they are not used by the UKCA code.

### Using python to regrid your emissions

For this task we will be using the Iris (http://scitools.org.uk/iris/) python library, which is developed by the Met Office and can read Met Office formatted files. The cf-python (http://cfpython.bitbucket.org/) library also does this, but we will not be using this here.

#### ARCHER

You can use Iris on ARCHER by loading the anaconda (https://www.continuum.io/anaconda-overview) module:

```
module load anaconda/2.2.0-python2
```

Note that this is not the default anaconda module, which does not contain Iris.

If you then

```
conda list
```

you will see that Iris is available:

```
iris                      1.10.0               np19py27_2
```

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_v10.9_Tutorial_5
Virtual Machine

To be able to use Iris on the VM, you should first run the command

```
install-iris
```

which installs the required python packages from anaconda (https://www.continuum.io/anaconda-overview), and also creates an alias in your ~/.bashrc file, called `conda`, which opens a terminal with additional paths set. You may need to re-source your ~/.bashrc by

```
~/.bashrc
```

before running the `conda` command.

Using python

When developing python scripts, a handy command to use can be set by the following alias:

```
alias pylab='ipython --pylab --logfile=ipython-`date +"%Y%m%d-%H%M%S"`.py'
```

This uses ipython (http://ipython.readthedocs.io/en/stable/), which is an interactive shell for python. The --pylab automatically loads some standard python libraries such as scipy (https://www.scipy.org/), numpy (http://www.numpy.org/), and matplotlib (http://matplotlib.org/). The --logfile means that all commands will be recorded in a file with the format `ipython-YYYYMMDD-HHMMSS.py`.

You can also execute pre-written python scripts by

- ARCHER: `python2.7 -c "execfile('name_of_script.py')"
- vm: `/home/vagrant/miniconda2/bin/python2.7 -c "execfile('name_of_script.py')"

from the command-line, or running the execfile command within ipython.

However, don't panic. You are not expected to know python to complete this tutorial. You will be provided with an example script to use that you only need to edit a few lines to get to work. You do not need to write a script from scratch, just read-through the provided script and try to understand what it does, and why.

Example python script

In the directory:

```
/work/n02/n02/ukca/Tutorial/vn10.9/Task5.1
```

you will find the file `regrid_ALICE_N48e.py`. You should take a copy of this file, and using your output .pa file from your suite, you should regrid the emissions into the N48 ENDGame grid. You will later use the resulting netCDF file in your suite.

This file looks like:

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

# This file is part of the UKCA Tutorials:
# http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_Tutorials_at_vn10.9
# Copyright (C) 2017  University of Cambridge
# This is free software: you can redistribute it and/or modify it under the
# terms of the GNU Lesser General Public License as published by the Free Software
# Foundation, either version 3 of the License, or (at your option) any later
# version.
# It is distributed in the hope that it will be useful, but WITHOUT ANY
# WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A
# PARTICULAR PURPOSE.  See the GNU Lesser General Public License for more details.
# You find a copy of the GNU Lesser General Public License at <http://www.gnu.org/licenses/>.
# Written by N. Luke Abraham 2017-11-15 <nla27@cam.ac.uk>
# To use this script on ARCHER you should first
#   module load anaconda/2.2.0-python2
# To be able to use ncdump you should fist
#   module load nco
```

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_5
# preamble
import iris
import numpy

# --- CHANGE THINGS BELOW THIS LINE TO WORK WITH YOUR FILES ETC. ---
# name of file containing an ENDGame grid, e.g. your model output
# NOTE: all the fields in the file should be on the same horizontal
# grid, as the field used MAY NOT be the first in order of STASH
grid_file='/work/n02/n02/ukca/Tutorial/vn10.9/sample_output/Task4.1/atmosa.pa19810901_00'

# name of emissions file
emissions_file='/work/n02/n02/ukca/Tutorial/vn10.9/Task5.1/Emissions_of_ALICE.nc'

# --- BELOW THIS LINE, NOTHING SHOULD NEED TO BE CHANGED ---

species_name='ALICE'

grid=iris.load_cube(grid_file,iris.AttributeConstraint(STASH='m01s34i010'))
grd.coord(axis='x').guess_bounds()
grd.coord(axis='y').guess_bounds()

# This is the original data
ems=iris.load_cube(emissions_file)

# make intersection between 0 and 360 longitude to ensure that the data is regridded correctly
nems = ems.intersection(longitude=(0, 360))

# make sure that we use the same coordinate system, otherwise regrid won't work
nems.coord(axis='x').coord_system=grd.coord_system()
nems.coord(axis='y').coord_system=grd.coord_system()

# now guess the bounds of the new grid prior to regridding
nems.coord(axis='x').guess_bounds()
nems.coord(axis='y').guess_bounds()

# now regrid
ocube=nems.regrid(grd,iris.analysis.AreaWeighted())

# now add correct attributes and names to netCDF file
ocube.var_name='emissions_'+str.strip(species_name)
ocube.long_name=str.strip(species_name)+' surf emissions'
ocube.units=cf_units.Unit('kg m-2 s-1')
ocube.attributes['vertical_scaling']='surface'
ocube.attributes['tracer_name']=str.strip(species_name)

# global attributes, so don’t set in local_keys
# NOTE: all these should be strings, including the numbers! This will change at a later UM version.
# basic emissions type
ocube.attributes['emission_type']='2' # periodic
ocube.attributes['update_type']='2'   # same as above
ocube.attributes['update_freq_in_hours']='120' # i.e. 5 days
ocube.attributes['source']='UKCA Tutorial Task 5.1 - creating netCDF emissions'

# rename and set time coord - set to be 0000/01/16:00:00-0000/12/16:00:00
# this bit is annoyingly fiddly
ocube.coord(axis='t').var_name='time'
ocube.coord(axis='t').standard_name='time'
ocube.coords(axis='t')[0].units=cf_units.Unit('hours since 1970-01-01', calendar='360_day')
ocube.coord(axis='t').points=numpy.array([-17020440, -17019720, -17019000, -17018280, 
-17017560, -17016840, -17016120, -17015400, 
-17014680, -17013960, -17013240, -17012520])

# make z-direction.
zdims=iris.coords.DimCoord(numpy.array([0]),standard_name = 'model_level_number',
                                      units='1',attributes={'positive':'up'})
ocube.add_aux_coord(zdims)
ocube=iris.util.new_axis(ocube, zdims)
# now transpose cube to put Z 2nd
ocube.transpose([1,0,2,3])

# make coordinates 64-bit
ocube.coord(axis='x').points=ocube.coord(axis='x').points.astype(dtype='float64')
ocube.coord(axis='y').points=ocube.coord(axis='y').points.astype(dtype='float64')

# for some reason, longitude_bounds are double, but latitude_bounds are float
ocube.coord('latitude').bounds=ocube.coord('latitude').bounds.astype(dtype='float64')
# add forecast_period & forecast_reference_time
# forecast_reference_time
frt=numpy.array([-17020080, -17019360, -17018640, -17017920,
                 -17017200, -17016480, -17015760, -17015040,
                 -17014320, -17013600, -17012880, -17012160],dtype='float64')
frt_dims=iris.coords.AuxCoord(frt,standard_name = 'forecast_reference_time',
                               units=cf_units.Unit('hours since 1970-01-01', calendar='360_day'))
ocube.add_aux_coord(frt_dims,data_dims=0)
ocube.coord('forecast_reference_time').guess_bounds()
# forecast_period
fp=numpy.array([-360],dtype='float64')
fp_dims=iris.coords.AuxCoord(fp,standard_name = 'forecast_period',
                           units=cf_units.Unit('hours'),bounds=numpy.array([-720,0],dtype='float64'))
ocube.add_aux_coord(fp_dims,data_dims=None)
# add-in cell_methods
ocube.cell_methods = [iris.coords.CellMethod('mean', 'time')]
# set _FillValue
fillval=1e+20
ocube.data = numpy.ma.array(data=ocube.data, fill_value=fillval, dtype='float32')
# output file name, based on species
outpath='ukca_emiss_'+species_name+'.nc'
# don't want time to be cattable, as is a periodic emissions file
iris.FUTURE.netcdf_no_unlimited=True
# annoying hack to set a missing_value attribute as well as a _FillValue attribute
dict.__setitem__(ocube.attributes, 'missing_value', fillval)
# now write-out to netCDF
saver = iris.fileformats.netcdf.Saver(filename=outpath, netcdf_format='NETCDF3_CLASSIC')
saver.update_global_attributes(Conventions=iris.fileformats.netcdf.CF_CONVENTIONS_VERSION)
saver.write(ocube, local_keys=['vertical_scaling', 'missing_value','um_stash_source','tracer_name'])
# end of script

# Why we are messing around with metadata?
#-----------------------------------------
# We need to adapt the metadata of the emissions data to
# match what UKCA is expecting.
# e.g. the metadata of the 'Emissions_of_ALICE.nc file is:
#
# netcdf Emissions_of_ALICE {
#    dimensions:
#            lon = 720 ;
#            lat = 360 ;
#            date = UNLIMITED ; // (12 currently)
#    variables:
#          float lon(lon) ;
#          lon:long_name = "Longitude" ;
#          lon:standard_name = "longitude" ;
#          lon:units = "degrees_east" ;
#          lon:point_spacing = "even" ;
#          lon:modulo = " " ;
#          float lat(lat) ;
#          lat:long_name = "Latitude" ;
#          lat:standard_name = "latitude" ;
#          lat:units = "degrees_north" ;
#          lat:point_spacing = "even" ;
#          float date(date) ;
#          date:long_name = "Time" ;
#          date:units = "days since 1960-01-01" ;
#          date:time_origin = "01-JAN-1960:00:00:00" ;
#          float ALICE(date, lat, lon) ;
#          ALICE:source = " " ;
#          ALICE:name = "ALICE" ;
#          ALICE:title = "Emissions of ALICE in kg/m^2/s" ;
#          ALICE:date = "01/01/60" ;
#          ALICE:time = "00:00" ;
#          ALICE:long_name = "Emissions of ALICE in kg/m^2/s" ;
#          ALICE:standard_name = "tendency_of_atmosphere_mass_content_of_ALICE_due_to_emission" ;
#          ALICE:units = "kg/m2/s" ;
#          ALICE:missing_value = 2.0e+20f ;
#          ALICE:_FillValue = 2.0e+20f ;
#          ALICE:valid_min = 0.f ;
#          ALICE:valid_max = 2.60646e-08f ;
#          // global attributes:
#          :history = "Tue Jun 18 14:32:42 BST 2013 - XCONV V1.92 16-February-2006" ;
#    }

whereas, the metadata of the
/file is, e.g.:

```{bash}
netcdf ukca_emiss_CO {
  dimensions:
  time = UNLIMITED ; // (12 currently)
  model_level_number = 1 ;
  latitude = 72 ;
  longitude = 96 ;
  bnds = 2 ;

  variables:
  double emissions_CO(time, model_level_number, latitude, longitude) ;
  emissions_CO:long_name = "CO surf emissions" ;
  emissions_CO:units = "kg m-2 s-1" ;
  emissions_CO:um_stash_source = "m01s00i303" ;
  emissions_CO:tracer_name = "CO" ;
  emissions_CO:vertical_scaling = "surface" ;
  emissions_CO:cell_methods = "time: mean" ;
  emissions_CO:grid_mapping = "latitude_longitude" ;
  emissions_CO:coordinates = "forecast_period forecast_reference_time" ;
  int latitude_longitude ;
  latitude_longitude:grid_mapping_name = "latitude_longitude" ;
  latitude_longitude:longitude_of_prime_meridian = 0. ;
  latitude_longitude:earth_radius = 6371229. ;
  double time(time) ;
  time:axis = "T" ;
  time:bounds = "time_bnds" ;
  time:units = "hours since 1970-01-01 00:00:00" ;
  time:standard_name = "time" ;
  time:calendar = "360_day" ;

  double time_bnds(time, bnds) ;
  int model_level_number(model_level_number) ;
  model_level_number:axis = "Z" ;
  model_level_number:units = "metre" ;
  model_level_number:standard_name = "model_level_number" ;
  model_level_number:long_name = "height at theta layer midpoint" ;
  model_level_number:positive = "up" ;

  float latitude(latitude) ;
  latitude:axis = "Y" ;
  latitude:bounds = "latitude_bnds" ;
  latitude:units = "degrees_north" ;
  latitude:standard_name = "latitude" ;

  float latitude_bnds(latitude, bnds) ;

  float longitude(longitude) ;
  longitude:axis = "X" ;
  longitude:bounds = "longitude_bnds" ;
  longitude:units = "degrees_east" ;
  longitude:standard_name = "longitude" ;

  double longitude_bnds(longitude, bnds) ;

  double forecast_period ;
  forecast_period:bounds = "forecast_period_bnds" ;
  forecast_period:units = "hours" ;
  forecast_period:standard_name = "forecast_period" ;

  double forecast_period_bnds(bnds) ;

  double forecast_reference_time(time) ;
  forecast_reference_time:units = "hours since 1970-01-01 00:00:00" ;
  forecast_reference_time:standard_name = "forecast_reference_time" ;
  forecast_reference_time:calendar = "360_day" ;

  // global attributes:
  :emission_type = "2" ;
  :source = "Data from Met Office Unified Model" ;
  :um_version = "7.3" ;
  :um_stash_in_hours = "120" ;
  :um_update_type = "2" ;
  :Conventions = "CF-1.5" ;
}
```

so the metadata of our new emissions file needs to be edited to be what UKCA
expects.

After using this script, the resultant netCDF file should look like:

```{bash}
netcdf ukca_emiss_ALICE {
  dimensions:
  time = 12 ;
  model_level_number = 1 ;
  latitude = 72 ;
  longitude = 96 ;

  // global attributes:
  :emission_type = "2" ;
  :source = "Data from Met Office Unified Model" ;
  :um_version = "7.3" ;
  :um_stash_in_hours = "120" ;
  :um_update_type = "2" ;
  :Conventions = "CF-1.5" ;
}
```

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_5
#                 bnds = 2 ;
#    variables:
#            float emissions_ALICE(time, model_level_number, latitude, longitude) ;
#                    emissions_ALICE:_FillValue = 1.e+20f ;
#                    emissions_ALICE:long_name = "ALICE surf emissions" ;
#                    emissions_ALICE:units = "kg m-2 s-1" ;
#                    emissions_ALICE:missing_value = 1.e+20 ;
#                    emissions_ALICE:tracer_name = "ALICE" ;
#                    emissions_ALICE:vertical_scaling = "surface" ;
#                    emissions_ALICE:cell_methods = "time: mean" ;
#                    emissions_ALICE:grid_mapping = "latitude_longitude" ;
#                    emissions_ALICE:coordinates = "forecast_period forecast_reference_time" ;
#            int latitude_longitude ;
#                    latitude_longitude:grid_mapping_name = "latitude_longitude" ;
#                    latitude_longitude:longitude_of_prime_meridian = 0. ;
#                    latitude_longitude:earth_radius = 6371229. ;
#            double time(time) ;
#                    time:axis = "T" ;
#                    time:units = "hours since 1970-01-01" ;
#                    time:standard_name = "time" ;
#                    time:long_name = "Time" ;
#                    time:calendar = "360_day" ;
#                    time:time_origin = "01-JAN-1960:00:00:00" ;
#            int model_level_number(model_level_number) ;
#                    model_level_number:axis = "Z" ;
#                    model_level_number:units = "1" ;
#                    model_level_number:standard_name = "model_level_number" ;
#                    model_level_number:positive = "up" ;
#            double latitude(latitude) ;
#                    latitude:axis = "Y" ;
#                    latitude:bounds = "latitude_bnds" ;
#                    latitude:units = "degrees_north" ;
#                    latitude:standard_name = "latitude" ;
#            double latitude_bnds(latitude, bnds) ;
#            double longitude(longitude) ;
#                    longitude:axis = "X" ;
#                    longitude:bounds = "longitude_bnds" ;
#                    longitude:units = "degrees_east" ;
#                    longitude:standard_name = "longitude" ;
#            double longitude_bnds(longitude, bnds) ;
#            double forecast_period ;
#                    forecast_period:bounds = "forecast_period_bnds" ;
#                    forecast_period:units = "hours" ;
#                    forecast_period:standard_name = "forecast_period" ;
#            double forecast_period_bnds(bnds) ;
#            double forecast_reference_time(time) ;
#                    forecast_reference_time:bounds = "forecast_reference_time_bnds" ;
#                    forecast_reference_time:units = "hours since 1970-01-01" ;
#                    forecast_reference_time:standard_name = "forecast_reference_time" ;
#                    forecast_reference_time:calendar = "360_day" ;
#            double forecast_reference_time_bnds(time, bnds) ;
#            double forecast_reference_time_bnds(bnds) ;
#                    forecast_reference_time:bounds = "forecast_reference_time_bnds" ;
#                    forecast_reference_time:units = "hours since 1970-01-01" ;
#                    forecast_reference_time:standard_name = "forecast_reference_time" ;
#                    forecast_reference_time:calendar = "360_day" ;
#            double forecast_reference_time_bnds(bnds) ;
#    // global attributes:
#                    :Conventions = "CF-1.5" ;
#                    :date = "01/01/60" ;
#                    :emission_type = "2" ;
#                    :history = "Tue Jun 18 14:32:42 BST 2013 - XCONV V1.92 16-February-2006" ;
#                    :invalid_standard_name = "tendency_of_atmosphere_mass_content_of_ALICE_due_to_emission" ;
#                    :name = "ALICE" ;
#                    :source = "UKCA Tutorial Task 5.1 - creating netCDF emissions" ;
#                    :time = "00:00" ;
#                    :title = "Emissions of ALICE in kg/m^2/s" ;
#                    :update_freq_in_hours = "120" ;
#                    :update_type = "2" ;
#                    :valid_max = 2.60646e-08f ;
#                    :valid_min = 0.f ;
# )

All scripts provided for these tutorials can also be found on GitHub (https://github.com/theabro/ukca).

Solution to Task5.1

You were given the task:

- In the

/work/n02/n02/ukca/Tutorial/vn10.9/Task5.1
directory on ARCHER there is the file `Emissions_of_ALICE.nc` which is a 0.5x0.5 degree resolution surface emission field. You should regrid this file to the N48 ENDGame grid, and output it as a netCDF file that has the required netCDF metadata for a climatological surface emission without any diurnal cycle.

It should be straightforward to point to your .pa file to use in the script. You should read through it and have an understanding of how it works.

Once the file has been regridded, you should see the output looking like this:

```
ALICE emissions prior to regridding.

ALICE emissions after regridding. Note the shift of the grid.
```

A file that has been produced by the above script can be found at

```
/work/n02/n02/ukca/Tutorial/vn10.9/Task5.1/solution/ukca_emiss_ALICE.nc
```

**Task 5.2: make the required code changes to add your emission into UKCA**

**TASK 5.2:** You should now make the UKCA code changes to add your emission into the ALICE tracer. Use STASH code s50i304 for the diagnostic.

**Hint**

You will need to add-in the molar mass of ALICE. You can calculate this from the mass of air and the conversion factor defined in Task 4.1. You will need to add code to allow a diagnostic of the emission to be output.

**Rose changes**

There are several ways that the location of emissions files can be defined within the UKCA panel. These are:

1. With a full path to the file
2. Using environment variables set using an `ancil_versions` file
3. A mixture of (1) and (2)
4. By specifying a top-level directory in the `ukca_em_dir` Rose variable, and then paths to the required files within `ukca_em_files`

Within Rose you may also find that the files are in separate boxes (one for each file) or in a single long list.

The UKCA training suite uses option 4 above, specifying both `ukca_em_dir` and `ukca_em_files`.

**ARCHER**

As `ukca_em_dir` is within `UN_INSTALL_DIR`, you will need to either copy or symbolically-link the directories from here to your own space on `/work`

`UN_INSTALL_DIR` is set in the `suite.rc` file to be `UMDIR`, so you can make the required changes like this:

```
mkdir ukca_emiss
cd ukca_emiss
ln -s /work/n02/n02/hum/ancil/atmos/n48e/ukca_emiss/andres_kasgnoc
ln -s /work/n02/n02/hum/ancil/atmos/n48e/ukca_emiss/cmip5
ln -s /work/n02/n02/hum/ancil/atmos/n48e/ukca_emiss/gfed3.1
```

You can then either copy or symbolically link your newly-created emissions file into this directory as well.

This directory will need to be on `/work` as the batch queues are unable to see `/home`.

**Virtual Machine**

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_5
On the VM you do have write permissions for UMDIR, which is located at /home/vagrant/umdir, so you could either copy your new emissions file there, or follow the ARCHER instructions above.

Make your Rose changes

Once you have created the required directory structure you should change the value of ukca_em_dir to point to your new directory, and then go to the ukca_em_files list. Click the blue plus symbol to add an extra line, and put the path to the new file, relative to this directory.

If your emissions files are defined with the full path, and/or in a single long list, make the changes accordingly.

Code changes

ukca_chem_master.F90
You can choose to put a 1 in the E column of the chch_defs array. This is just for completeness however, as this column is not currently used.

ukca_setd1defs.F90
You will need to edit the em_chem_spec array for the scheme that you are using to include the new species that you are emitting into. By convention, these are ordered as 2D fields followed by 3D fields, although this order is unimportant other than the fact that the ‘NO_aircrft’ entry MUST be last in the list.
If the tracer is valid for both with and without aerosol chemistry (& GLOMAP) you will need to put this in twice. Also remember to increment this size of the em_chem_spec array, given by n_chem_emissions variable.

ukca_constants.F90
You will need to define the M_species for the emitted species. This should be consistent with the C_species value set in the previous tutorial.

get_molmass_mod.F90
You will need to add to the species_name CASE statement, to include a line such as this

```
CASE ('ALICE   ')
  get_molmass = m_ALICE
```

for each new species that you are emitting into.

Emissions diagnostics

By default, when emitting into a new species, the code requires you to create a diagnostic for these emissions as well. This is done by considering the following file and routines:

STASHmaster_A and STASHmaster-meta.conf

You should add a diagnostic in a similar way to the section 34 tracers, but in this case to section 50. It is best to copy an existing 2D (or 3D) emissions diagnostic specification, and make only the required edits to that.

Once you have made these code changes you should commit your branch to ensure that these changes are picked-up by your suite correctly.

get_emdiag_stash_mod.F90

Once you have decided on the STASH code to use, you need to add the species and 3-digit item code to this routine in the CASE statement.

ukca_emdiags_struct_mod.F90

This routine contains the derived type that holds the emissions fluxes for all species. You will need to add a logical (used to determine whether the output has been requested in STASH) and a pointer to a 2D or 3D array that will hold the emissions flux.

ukca_update_emdiagsstruct_mod.F90

In this routine you will need to add the initialisation of the logical added in ukca_emdiags_struct_mod.F90, as well as a small block of code that copies the 2D/3D em_diags data into the newly created pointer in the derived type. The logical will also need to be set to .TRUE. if STASH output has been requested (determined by the sf(item,section) look-up array).

Here it is easiest to copy an existing code-block and make the necessary changes for your new emission.

ukca_emiss_diags_mod.F90

In this routine you will add a unique code-block for this diagnostic that copies the pointer in the derived type into the stashwork array using the copydiag copydiag_3d routines. The stashwork array contains all the fields that will be output through STASH.

Here it is easiest to copy an existing code-block and make the necessary changes for your new emission.

ukca_emiss_ctl_mod.F90

The call to ukca_emiss_diags is protected by an IF statement, and you'll need to add a check to the entry in the sf array to see if the diagnostic is on. This should be of the format sf(N,section), where N is the item number of your new emissions diagnostic.

Solution to Task 5.2

You were given the task

- You should now make the UKCA code changes to add your emission into the ALICE tracer. Use STASH code s50304 for the diagnostic.
You were given the hints:

- You will need to add-in the molar mass of ALICE. You can calculate this from the mass of air and the conversion factor defined in Task 4.1.
- You will need to add code to allow a diagnostic of the emission to be output.

Once you have made the required code and Rose changes, you should now see that you ALICE tracer (s34i064) no longer contains noise, but now looks a bit like the emissions field you created in Task 5.1.

For a working Rose suite that has completed this task, please see:

**ARCHER:**

- **u-as292859922**
- **vm:** u-as297859923

The specific Rose changes made are:

- **ARCHER:** https://code.metoffice.gov.uk/trac/rooses-u/changeset/59922/a/s/2/9/2/trunk
- **vm:**
  - https://code.metoffice.gov.uk/trac/rooses-u/changeset/59921/a/s/2/9/7/trunk
  - https://code.metoffice.gov.uk/trac/rooses-u/changeset/59923/a/s/2/9/7/trunk

The specific Rose changes made are:

**ARCHER:**

Index: app/um/rose-app.conf
---
+++     app/um/rose-app.conf (revision 59818)
@@ -2641,7 +2641,7 @@
-ukca_em_dir='$UMDIR/ancil/atmos/n48e/ukca_emiss'
+ukca_em_dir='/home/vagrant/ukca_emiss'
media_kasgnoc/v1/ukca_emiss_SO2_nat.nc',
+ 'Task5.1/solution/ukca_emiss_ALICE.nc'
ukca_emiss_ALICE_OK=1

**ARCHER:**

Index: app/um/rose-app.conf
---
+++     app/um/rose-app.conf (revision 59818)
@@ -2638,7 +2638,7 @@
-ukca_em_dir='$UM_INSTALL_DIR/ancil/atmos/n48e/ukca_emiss'
+ukca_em_dir='/work/n02/n02/ukca/Tutorial/vn10.9/ukca_emiss'
media_kasgnoc/v1/ukca_emiss_SO2_nat.L38.nc',
+ 'Task5.1/solution/ukca_emiss_ALICE.nc'
ukca_emiss_ALICE_OK=1

**vm:**

Index: app/um/rose-app.conf
---
+++     app/um/rose-app.conf (revision 59815)
@@ -2641,7 +2641,7 @@
-ukca_em_dir='$UM_INSTALL_DIR/ancil/atmos/n48e/ukca_emiss'
+ukca_em_dir='/home/ukca/Tutorial/vn10.9/ukca_emiss'
media_kasgnoc/v1/ukca_emiss_SO2_nat.L38.nc',
+ 'Task5.1/solution/ukca_emiss_ALICE.nc'

These differences can be found in the file /home/ukca/Tutorial/vn10.9/worked_solutions/Task5.2/Task5.2_rose.patch on PUMA.

---

You will need to add-in the molar mass of ALICE. You can calculate this from the mass of air and the conversion factor defined in Task 4.1.

You will need to add code to allow a diagnostic of the emission to be output.

Once you have made the required code and Rose changes, you should now see that you ALICE tracer (s34i064) no longer contains noise, but now looks a bit like the emissions field you created in Task 5.1.

For a working Rose suite that has completed this task, please see:

**ARCHER:**

- **u-as292859922**
- **vm:** u-as297859923

The specific Rose changes made are:

- **ARCHER:** https://code.metoffice.gov.uk/trac/rooses-u/changeset/59922/a/s/2/9/2/trunk
- **vm:**
  - https://code.metoffice.gov.uk/trac/rooses-u/changeset/59921/a/s/2/9/7/trunk
  - https://code.metoffice.gov.uk/trac/rooses-u/changeset/59923/a/s/2/9/7/trunk

The specific Rose changes made are:

**ARCHER:**

Index: app/um/rose-app.conf
---
+++     app/um/rose-app.conf (revision 59818)
@@ -2638,7 +2638,7 @@
-ukca_em_dir='$UM_INSTALL_DIR/ancil/atmos/n48e/ukca_emiss'
+ukca_em_dir='/work/n02/n02/ukca/Tutorial/vn10.9/ukca_emiss'
media_kasgnoc/v1/ukca_emiss_SO2_nat.L38.nc',
+ 'Task5.1/solution/ukca_emiss_ALICE.nc'
ukca_emiss_ALICE_OK=1

**vm:**

Index: app/um/rose-app.conf
---
+++     app/um/rose-app.conf (revision 59815)
@@ -2641,7 +2641,7 @@
-ukca_em_dir='$UM_INSTALL_DIR/ancil/atmos/n48e/ukca_emiss'
+ukca_em_dir='/home/ukca/Tutorial/vn10.9/ukca_emiss'
media_kasgnoc/v1/ukca_emiss_SO2_nat.L38.nc',
+ 'Task5.1/solution/ukca_emiss_ALICE.nc'
ukca_emiss_ALICE_OK=1

---

You were given the hints:

- You will need to add-in the molar mass of ALICE. You can calculate this from the mass of air and the conversion factor defined in Task 4.1.
- You will need to add code to allow a diagnostic of the emission to be output.

Once you have made the required code and Rose changes, you should now see that you ALICE tracer (s34i064) no longer contains noise, but now looks a bit like the emissions field you created in Task 5.1.

For a working Rose suite that has completed this task, please see:

**ARCHER:**

- **u-as292859922**
- **vm:** u-as297859923

The specific Rose changes made are:

- **ARCHER:** https://code.metoffice.gov.uk/trac/rooses-u/changeset/59922/a/s/2/9/2/trunk
- **vm:**
  - https://code.metoffice.gov.uk/trac/rooses-u/changeset/59921/a/s/2/9/7/trunk
  - https://code.metoffice.gov.uk/trac/rooses-u/changeset/59923/a/s/2/9/7/trunk

The specific Rose changes made are:

**ARCHER:**

Index: app/um/rose-app.conf
---
+++     app/um/rose-app.conf (revision 59818)
@@ -2638,7 +2638,7 @@
-ukca_em_dir='$UM_INSTALL_DIR/ancil/atmos/n48e/ukca_emiss'
+ukca_em_dir='/work/n02/n02/ukca/Tutorial/vn10.9/ukca_emiss'
media_kasgnoc/v1/ukca_emiss_SO2_nat.L38.nc',
+ 'Task5.1/solution/ukca_emiss_ALICE.nc'
ukca_emiss_ALICE_OK=1

**vm:**

Index: app/um/rose-app.conf
---
+++     app/um/rose-app.conf (revision 59815)
@@ -2641,7 +2641,7 @@
-ukca_em_dir='$UM_INSTALL_DIR/ancil/atmos/n48e/ukca_emiss'
+ukca_em_dir='/home/ukca/Tutorial/vn10.9/ukca_emiss'
media_kasgnoc/v1/ukca_emiss_SO2_nat.L38.nc',
+ 'Task5.1/solution/ukca_emiss_ALICE.nc'
ukca_emiss_ALICE_OK=1

---
ukca_h1202mmr=3.788e-13
ukca_h1211mmr=2.225e-11
ukca_h1301mmr=1.363e-11

For a working UM branch that has completed this task, please see
The specific UM changes made are:

```
Index: src/atmosphere/UKCA/ukca_setd1defs.F90
===================================================================
--- src/atmosphere/UKCA/ukca_setd1defs.F90 (revision 46535)
+++ src/atmosphere/UKCA/ukca_setd1defs.F90 (revision 46574)
@@ -313,7 +313,7 @@
   crs_clt(31) = 'ALICE'
   crs_clt(32) = 'NO_aircraft'

-  c5h8 = 9
+  c5h8 = 10
   
```

```
Index: src/atmosphere/UKCA/ukca_chem_master.F90
===================================================================
--- src/atmosphere/UKCA/ukca_chem_master.F90 (revision 46535)
+++ src/atmosphere/UKCA/ukca_chem_master.F90 (revision 46574)
@@ -377,7 +377,7 @@
   chch_t1(113,'MEMALD    ',1,'TR        ','          ',0,0,0,0,107),&
   chch_t1(114,'GLY       ',1,'TR        ','          ',0,1,0,0,107),&
   chch_t1(115,'oXYLENE   ',1,'TR        ','          ',0,0,1,0,107),&
-  chch_t1(116,'ALICE     ',1,'TR        ','          ',0,0,0,ST,107),&
+  chch_t1(116,'ALICE     ',1,'TR        ','          ',0,0,1,ST,107),&
   chch_t1(117,'BOB       ',1,'TR        ','          ',0,0,0,0,107)/

! Heterogeneous chemistry
Index: src/atmosphere/UKCA/get_molmass_mod.F90
===================================================================
--- src/atmosphere/UKCA/get_molmass_mod.F90 (revision 46535)
+++ src/atmosphere/UKCA/get_molmass_mod.F90 (revision 46574)
@@ -243,6 +243,11 @@
```

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_5
get_molmass = m_brcl

! -----------------------------------------
+ ! UKCA Tutorial Tracer
+CASE ('ALICE     '
+  get_molmass = m_ALICE
+
+ ! Others (report warning)
CASE ('AGE      '
  get_molmass = 1.0
Index: src/atmosphere/UKCA/ukca_constants.F90
===================================================================
--- src/atmosphere/UKCA/ukca_constants.F90 (revision 46535)
+++ src/atmosphere/UKCA/ukca_constants.F90 (revision 46574)
@@ -370,6 +370,8 @@
REAL, PARAMETER :: m_mgly    = 72.0
REAL, PARAMETER :: m_hacet  = 74.0
+!     UKCA Tutorial tracers - only ALICE is emitted
+REAL, PARAMETER :: m_ALICE    = 28.97
!     Extra masses for EXTTC chemistry
REAL, PARAMETER :: m_apin     = 136.0
Index: src/atmosphere/UKCA/ukca_emiss_diags_mod.F90
===================================================================
--- src/atmosphere/UKCA/ukca_emiss_diags_mod.F90 (revision 46535)
+++ src/atmosphere/UKCA/ukca_emiss_diags_mod.F90 (revision 46574)
@@ -584,6 +584,23 @@
END IF
END IF

+CASE ('ALICE     '
  get_emdiag_stash = 304
+CASE DEFAULT
! Report error unless this is an aerosol emission (BC_fossil:, etc)
! No diagnostics for these source emissions because the resulting

Index: src/atmosphere/UKCA/ukca_emiss_ctl_mod.F90
===================================================================
--- src/atmosphere/UKCA/ukca_emiss_ctl_mod.F90 (revision 46535)
+++ src/atmosphere/UKCA/ukca_emiss_ctl_mod.F90 (revision 46574)
@@ -788,7 +788,8 @@
 sf(168,section) .OR. sf(169,section) .OR. sf(170,section) .OR.      &
 sf(171,section) .OR. sf(172,section) .OR. sf(211,section) .OR. &
 sf(212,section) .OR. sf(213,section) .OR. sf(214,section) .OR. &
 sf(215,section) .OR. sf(216,section) .OR. sf(217,section) ) THEN
+ sf(215,section) .OR. sf(216,section) .OR. sf(217,section) .OR.      &
+ sf(304,section) THEN
 CALL ukca_emiss_diags (row_length, rows, model_levels,                &
 len_stashwork50, stashwork50)
Index: src/atmosphere/UKCA/get_emdiag_stash_mod.F90
===================================================================
--- src/atmosphere/UKCA/get_emdiag_stash_mod.F90 (revision 46535)
+++ src/atmosphere/UKCA/get_emdiag_stash_mod.F90 (revision 46574)
@@ -142,6 +142,9 @@
 CASE ('SO2_nat   ')  
get_emdiag_stash = 217
+CASE ('ALICE     ')  
get_emdiag_stash = 304
+CASE DEFAULT
! Report error unless this is an aerosol emission (BC_fossil:, etc)
! No diagnostics for these source emissions because the resulting

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_5
```fortran
LOGICAL :: l_em_so2low
LOGICAL :: l_em_so2hi
LOGICAL :: l_em_so2nat
LOGICAL :: l_em_ALICE

! Pointers to hold emission diagnostics
REAL, POINTER :: em_no     (:,:)
REAL, POINTER :: em_so2low (:,:)
REAL, POINTER :: em_so2hi  (:,:)
REAL, POINTER :: em_so2nat (:,:,:)
REAL, POINTER :: em_ALICE  (:,:),

END TYPE emdiags_struct
```

```fortran
emdiags%l_em_so2low = .FALSE.
emdiags%l_em_so2hi  = .FALSE.
emdiags%l_em_so2nat = .FALSE.
emdiags%l_em_ALICE  = .FALSE.
```

```fortran
CASE ('ALICE      ')
  IF (emdiags%l_em_ALICE) THEN
    emdiags%em_ALICE (:,:) = em_diags (:,:,1)
  ELSE
    IF (sf(item,section)) THEN
      ALLOCATE (emdiags%em_ALICE (row_length, rows))
      emdiags%em_ALICE (:,:) = em_diags (:,:,1)
      emdiags%l_em_ALICE     = .TRUE.
    END IF
  END IF
END IF
```

```fortran
# Available if chemistry is turned on in UKCA

[stashmaster:code(50304)]
description=alice emission flux (kg m-2 s-1)
help=Diagnostic to check the total ALICE surface emission flux

[stashmaster:code(51001)]
description=O3 MASS MIXING RATIO ON PRESS LEVS
help=Ozone Mass Mixing Ratio in kg/kg(Air)
```

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_5
These differences can be found in the file /home/ukca/Tutorial/vn10.9/worked_solutions/Task5.2/Task5.2_code.patch on PUMA.

Sample output from this task can be found at /work/n02/n02/ukca/Tutorial/vn10.9/sample_output/Task5.2/atmosa.pal9810901_00 on ARCHER.

**Task 5.3: Output the diagnostic of your new emission**

**Task 5.3:** Now that you are able to run the model with your new emission file, output this as a diagnostic to the UPA stream as a 3-hour mean.

You should reference the what is STASH tutorial for information on how to do this

**Solution to Task 5.3**

You were given the task:

- Now that you are able to run the model with your new emission file, output this as a diagnostic to the UPA stream as a 3-hour mean.

This achieved by outputting s50304 to the UPA stream in STASH.

You will now find the additional field in your output file:

```
23 : 96  72  1  1  mflux: Stash code 50304
```

For a working Rose suite that has completed this task, please see

- **ARCHER:** u-as292859937
- **vm:** u-as297859931

The specific Rose changes made are:

- **ARCHER:** https://code.metoffice.gov.uk/trac/roses-u/changeset/59937/a/s/2/9/2/trunk
- **vm:** https://code.metoffice.gov.uk/trac/roses-u/changeset/59931/a/s/2/9/7/trunk

The specific Rose changes made are:

**ARCHER:**

```
Index: app/um/rose-app.conf
===================================================================
--- app/um/rose-app.conf (revision 59922)
+++ app/um/rose-app.conf (revision 59937)
@@ -3449,6 +3449,14 @@
     dom_name='DIAG'
     *isec=50
     *item=304
+
   *tim_name='T3HNN'
   *use_name='UPA'
   *[namelist:umstash_streq(50304_f1d61e7a)]
+
   tim_name='T3HNN'
   use_name='UPA'
   *[namelist:umstash_streq(51001_3e6241a4)]
```

**vm:**

```
Index: app/um/rose-app.conf
===================================================================
--- app/um/rose-app.conf (revision 59923)
+++ app/um/rose-app.conf (revision 59931)
@@ -4041,6 +4041,14 @@
     dom_name='DIAG'
     *isec=50
     *item=304
+
   *tim_name='T3HNN'
   *use_name='UPA'
   *[namelist:umstash_streq(50304_f1d61e7a)]
+
   tim_name='T3HNN'
   use_name='UPA'
   *[namelist:umstash_streq(51001_3e6241a4)]
```

These differences can be found in the file /home/ukca/Tutorial/vn10.9/worked_solutions/Task5.3/Task5.3_rose.patch on PUMA.

ALICE emission flux diagnostic output.
Sample output from this task can be found at /work/n02/n02/ukca/Tutorial/vn10.9/sample_output/Task5.3/atmosa.pa.19810901_00 on ARCHER.

Checklist

☐ Obtain emissions data for the species of interest.
☐ Regrid the emissions to the correct MetUM resolution that you are using.
☐ Save these emissions as netCDF, including the required metadata that UKCA requires.
☐ In your suite, include this file in the ukca_em_files variable in the UKCA panel: um → namelist → UM Science Settings → Section 34 - UKCA: UK Aerosols and Chemistry.
☐ (Optional) Put a 1 in the E column of the chch_defs_master in ukca_chem_master.F90.
☐ Put the species being emitted into the em_chem_spec array in ukca_setd1defs.F90, and increment n_chem_emissions.
☐ Put the correct M_species value in ukca_constants.F90.
☐ In get_molmass_mod.F90, append the CASE statement with your new species.
☐ Make diagnostics output slots for your new emissions in STASHmaster_A and add help-text to STASHmaster-meta.conf.
☐ In get_emdiag_stash_mod.F90 add the species and 3-digit item code to the CASE statement.
☐ In ukca_emdiags_struct_mod.F90 add a logical and array pointer to the derived type for your new species.
☐ In ukca_update_emdiagsstruct_mod.F90 initialise your new derived type entries, and add a block of code to copy the emissions field into the diagnostic pointer.
☐ In ukca_emiss_diags_mod.F90 add a code block to copy the field to the stashwork array using copydiag/copydiag_3d.
☐ In ukca_emiss_ctl_mod.F90 add the sf(ITEM,section) check in the IF statement protecting the call to ukca_emiss_diags.
☐ fcm commit the changes to your branch.
☐ Output your emissions diagnostic in: um → namelist → Model Input and Output → STASH Requests and Profiles → STASH Requests.
☐ Run the TidyStashTransform transform macro.
☐ Save your suite.
☐ In the roses/[SUITE-ID] directory, run fcm commit to commit your changes to the repository.
☐ Run your suite.

Tutorial 6

Written by Luke Abraham 2017. Many thanks to Alistair Sellar for the notes on netCDF metadata.


- This page was last modified on 9 January 2018, at 15:42.
UKCA Chemistry and Aerosol vn10.9 Tutorial 6

From UKCA

UKCA Chemistry and Aerosol Tutorials at vn10.9

Contents

1. What you will learn in this tutorial
2. Task 6.1: Add a bimolecular reaction
3. Adding new Chemical Reactions
4. Biomolecular Reactions
5. Termolecular Reactions
6. Heterogeneous Reactions
7. Photolysis Reactions
8. Solution to Task 6.1: Add a bimolecular reaction
9. Checklist

What you will learn in this tutorial

During this tutorial you will learn how UKCA specifies different chemical reactions. You will then add a new reaction involving the new tracers that you have added.

Task 6.1: Add a bimolecular reaction

**TASK 6.1:** You should now add in the bimolecular reaction of ALICE with OH to form BOB and a secondary organic compound (labelled in UKCA as Sec_Org). This reaction is given by:

\[
\text{ALICE} + \text{OH} \rightarrow \text{BOB} + \text{Sec}_\text{Org}
\]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k_0 )</td>
<td>2.70E-11</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>0.00</td>
</tr>
<tr>
<td>( \beta )</td>
<td>-390.00</td>
</tr>
</tbody>
</table>

Adding new Chemical Reactions

UKCA currently uses two different methods of defining the chemical reactions solved in the model. The first is a backward Euler solver, and is used for the RAQ and StdTrop chemistry schemes where the solver itself is created by a code-writer. The second makes use of the ASAD chemical integration software package ([http://www.atm.ch.cam.ac.uk/acmsu/asad/](http://www.atm.ch.cam.ac.uk/acmsu/asad/)), and is used for the CheT/TropIsop, CheS/Strat, and CheST/StratTrop chemistry schemes. ASAD can use many different solvers, although currently it uses a symbolic Newton-Raphson solver. In this tutorial we will only consider the ASAD framework, as this is easily extended by a user.

ASAD considers four different types of chemical reactions: bimolecular reactions, termolecular reactions, heterogeneous reactions, and photolysis reactions. To make changes and add reactions you will need to make changes to the UKCA source code which can be found in

```
vn10.9_your_branch_name/src/atmosphere/UKCA
```

During this tutorial you will be tasked with adding a new reaction into your branch.

Biomolecular Reactions

For most bimolecular reactions, it is sufficient to provide the \( k_0 \), \( \alpha \), and \( \beta \) coefficients that are used to compute the rate coefficient \( k \) from the Arrhenius expression

\[
k = k_0 \left( \frac{T}{300} \right)^\alpha \exp \left( \frac{-\beta}{T} \right)
\]

Bimolecular Reaction Definition

The bimolecular reactions are defined in the `ukca_chem_master.F90` module using the `ratb_t1` Fortran type specification, and are held in arrays. At the end of this routine the `ratb_defs_master` array is created from these, and if that scheme is selected in UKCA these reactions are copied across into the master `ratb_defs` array.

The format of this `ratb_t1` type (defined in `ukca_chem_defs_mod.F90`) is
If fractional products are not required for a reaction, then the fraction of each product formed should be set to 0.00. If fractional products are required for any one of the products then the fraction of each product formed should be set to its correct value.

The settings for \( k \), \( \alpha \), \( \beta \), Fraction of Product 1 produced, Fraction of Product 2 produced, Fraction of Product 3 produced, Fraction of Product 4 produced, SCHEME, QUALIFIER, DISQUALIFIER, VN) are the same as in the adding new tracers tutorial, although here \( k \) should be incremented for each new reaction, where there might be the same reaction specified several times with changes to reaction rates or even species.

The first reaction in these examples takes its kinetic data from IUPAC (http://www.iupac-kinetic.ch.cam.ac.uk/). Going to this website, this reaction is defined here (http://www.iupac-kinetic.ch.cam.ac.uk/datasheets/xhtml/HOx_VOC8_HO_CH2C%28CH3%29CHCH2%28isoprene%29.xhtml_mathml.xml). The second and third reaction above take their kinetic data from NASA’s Jet Propulsion Laboratory (http://jpldataeval.jpl.nasa.gov/). The rate for this can be found on page 1-19 of the JPL2011 document (http://jpldataeval.jpl.nasa.gov/pdf/JPL%2010-6%20Final%2015June2011.pdf). You can see that in the instances above, different chemistry schemes use slightly different rates or species, as may be required by the scheme and species considered.

When adding new reactions you will need to increment the size of the array holding the ratb_t1 type.

To add new bimolecular reactions you will need to append equivalent lines for the new reactions to the end of the ratb_defs_master array (increasing the array sizes accordingly). If there is a reaction that is an exception to the general Arrhenius equation then special code needs to be placed in the asad_bimol.F90 routine, which is held in the UKCA/source-code directory.

**Termolecular Reactions**

As well as defining reactions involving a third body, the termolecular rate definition can also be used to define unimolecular reactions.

The pressure and temperature dependent rate, \( k \), of a termolecular reaction is given by

\[
k = \left( \frac{k_0 [M]}{1 + k_0 [M] / k_{\infty}} \right) F_c \left( 1 + \log_{10} \left( \frac{k_0 [M]}{k_{\infty}} \right) \right)^{-1}
\]

where the low pressure rate constant \( k_0 \), is given by

\[
k_0 = k_1 \left( \frac{T}{300} \right)^{\alpha_1} \exp \left( -\frac{\beta_1}{T} \right)
\]

and the high pressure rate constant \( k_{\infty} \) is given by

\[
k_{\infty} = k_2 \left( \frac{T}{300} \right)^{\alpha_2} \exp \left( -\frac{\beta_2}{T} \right)
\]

**Termolecular Reaction Definition**

The termolecular reactions are defined in the ukca_chem_master.F90 module using the ratb_t1 Fortran type specification and are held in the ratb_defs_master array.

To format of this ratb_t1 type is
and as in `rath_t1`, where the fraction of a product should be set to 0.0 if this functionality does not need to be used.

The settings for \( N \), `SCHEME`, `QUALIFIER`, `DISQUALIFIER`, and `Vn` are the same as in the adding new tracers tutorial, although here \( N \) should be incremented for each new reaction, where there might be the same reaction specified several times with changes to reaction rates or even species.

The \( f \) value is used to define the \( F_i \) value by

\[
F_i = \exp \left( -\frac{T}{f} \right)
\]

as \( F_i \) may or may not be highly temperature dependent.

Examples of these reactions are

```fortran
rath_t(N,'Reactant 1','Reactant 2','Product 1','Product 2',f, &
\( a_1 \), \( a_2 \), \( \beta_1 \), \( \beta_2 \), Fraction of Product 1 produced, Fraction of Product 2 produced,SCHEME,QUALIFIER, DISQUALIFIER,Vn), &
```

To add new heterogeneous reactions you will need to append equivalent lines for the new reactions to the end of the `rath_defs_master` array (increasing the array sizes accordingly). If there is any special code that needs to be added, this should be placed in the `asad_hetero.F90` routine, which is held in the UKCA/ source-code directory.

### Heterogeneous Reactions

Heterogeneous reactions are those that occur on aerosol surfaces. There is no functional form defined for these reactions, with special code needed to be added for each case.

#### Heterogeneous Reaction Definition

The heterogeneous reactions are defined in the `ukca_chem_master.F90` module using the `rath_t1` Fortran type specification, usually in one array (``rath_defs_master``).

To format of this `rath_t1` type is

```fortran
rath_t(N,'Reactant 1','Reactant 2','Product 1','Product 2','Product 3', &
\'Product 4\', Fraction of Product 1 produced, Fraction of Product 2 produced, Fraction of Product 3 produced, Fraction of Product 4 produced,SCHEME,QUALIFIER,DISQUALIFIER,Vn), &
```

i.e. there is no rate information provided. For reactions on PSCs special code has been added to the routines in `ukca_hetero_mod.F90`, and for other reactions there is code in `asad_hetero.F90`.

The settings for \( N \), `SCHEME`, `QUALIFIER`, `DISQUALIFIER`, and `Vn` are the same as in the adding new tracers tutorial, although here \( N \) should be incremented for each new reaction, where there might be the same reaction specified several times with changes to reaction rates or even species.

Examples of this type are

```fortran
rath_t(N,'Reactant 1','Reactant 2','Product 1','Product 2','Product 3', &
\'Product 4\', Fraction of Product 1 produced, Fraction of Product 2 produced, Fraction of Product 3 produced, Fraction of Product 4 produced,SCHEME,QUALIFIER,DISQUALIFIER,Vn), &
```

To add new heterogeneous reactions you will need to append equivalent lines for the new reactions to the end of the array (increasing the array sizes accordingly), before adding code to either `ukca_hetero_mod.F90` (for stratospheric reactions) or `asad_hetero.F90` (for tropospheric reactions).

### Photolysis Reactions

These define a reaction where a chemical compound is broken down by photons. There is no functional form defined for this type of reaction. Instead, either (in the troposphere) input files are used to define the reaction rates for each species, while (in the stratosphere) on-line look-up tables are generated for the rates for each species, or a separate photolysis code, `Fast-JX`, is used to interactively calculate the rate of reaction throughout the the whole atmosphere (for Fast-JX). These interactive schemes are preferred as they take the effect of aerosols or clouds into account at each timestep, allowing for more feedbacks to be investigated. In the upper stratosphere there are some wavelength regions that Fast-JX does not consider, and so the 3D on-line look-up tables are also used for these regions.

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_6
Tropospheric Off-Line Photolysis

If Fast-JX is not being used, then the off-line two-dimensional (zonally average) tropospheric photolysis is used (for all schemes). It is based on the work of Hough (1988)[1] and Law et al. (1998)[2].

This scheme makes use of datafiles which define the reaction rate for a particular species (e.g. H2O2), or if no rate is known, a nil rate can be used. For vn10.4 these files can be found in

```
$UMDIR/vn10.4/ctldata/UKCA/tropdata/photol
```

To use this scheme set the value of `i_ukca_photol` by clicking **2D Photolysis Scheme**. You will then need to give the location of the files (above). The code controlling this scheme is held in `ukca_phot2d.F90`.

It is advised that this scheme is no longer used, and Fast-JX interactive photolysis should be used instead.

**References**


Stratospheric Look-Up Table Photolysis

In a chemistry scheme which has stratospheric chemistry, such as CheS/Strat and CheST/StratTrop, if interactive photolysis is not used, then above 300hPa the look-up table approach of Lary and Pyle (1991)[1] is used (below 300hPa the tropospheric scheme described above is used). To use this scheme set the value of `i_ukca_photol` by clicking **2D Photolysis Scheme**. The code for this scheme is held in `ukca_strat_update.F90`.

**References**


Interactive Photolysis

The original Fast-J scheme (Wild et al, 2000)[1] uses 7 different wavelength bins appropriate for the troposphere, and the updated Fast-JX scheme (Neu et al., 2007)[2] adds up to an extra 11 bins allowing use in the stratosphere. At vn10.4 only Fast-JX is available, although previous UM version used Fast-J as well.

To use this scheme set the value of `i_ukca_photol` by clicking **Fast-JX**. You will then need to give the location of several input data files used by this scheme.

Further details on how the the Fast-JX scheme is used in UKCA, can be found in Telford et al (2013) (http://www.geosci-model-dev.net/6/161/2013/gmd-6-161-2013.html)[3].

The Fast-JX data files are found in

```
$UMDIR/vn10.9/ctldata/UKCA/fastj
```

on ARCHER.

**References**


Photolysis Reaction Definition

The photolysis reactions are defined in the `ukca_chem_master.F90` module using the `ratj_t1` Fortran type specification and held in the `ratj_defs_master` array.

To format this `ratj_t1` type is

```
ratj_t1(N,'Reactant 1','Reactant 2','Product 1','Product 2','Product 3',&
 'Product 4', Fraction of Product 1 produced, Fraction of Product 2 produced, Fraction of Product 3 produced, Fraction of Product 4 produced, Quantum Yield, Look-up Label,SCHEME,QUALIFIER,DISQUALIFIER,VN), &
```

The **Look-Up Label** is used to define the file used for the 2D photolysis, and is used by Fast-JX to find the correct values for each species in the input data files.

This is a 10-character string, although only the first 7 characters are read by Fast-JX. Reactant 2 will always be PHOTON.

The settings for `N`, `SCHEME`, `QUALIFIER`, `DISQUALIFIER`, and `VN` are the same as in the adding new tracers tutorial, although here `N` should be incremented for each new reaction, where there might be the same reaction specified several times with changes to reaction rates or even species.

**Examples of this type** are

```
3
3 This should produce H+ CHO -> H + HO2 + CO in ST scheme.
ratj_t1(3,'HCHO','PHOTON','H','HO2','CO','&
 '0,0,0,0,0,0,0,0,100.000','jhchoa','TI+T+ST+R,0,0,0,107) ,&
ratj_t1(3,'HCHO','PHOTON','H','CO','HO2','&
 '0,0,0,0,0,0,0,0,100.000','jhchoa','S,0,0,0,107) ,&
```

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_6
Solution to Task 6.1: Add a bimolecular reaction

You were given the task

- You should now add in the bimolecular reaction of ALICE with OH to form BOB and a secondary organic compound (labelled in UKCA as Sec.Org). This reaction is given by:

\[
\text{ALICE} + \text{OH} \rightarrow \text{BOB} + \text{Sec.Org}
\]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k_0 )</td>
<td>2.70E-11</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>0.00</td>
</tr>
<tr>
<td>( \beta )</td>
<td>-390.00</td>
</tr>
</tbody>
</table>

For a working Rose suite that has completed this task, please see

- ARCHER: u-as292860067
- vm: u-as297859986

The specific Rose changes made are:

- ARCHER: https://code.metoffice.gov.uk/trac/roses-u/changeset/60067/a/s/2/9/2/trunk
- vm: https://code.metoffice.gov.uk/trac/roses-u/changeset/59986/a/s/2/9/7/trunk

The specific Rose changes made are:

**ARCHER:**

Index: app/fcm_make/rose-app.conf

--- app/fcm_make/rose-app.conf (revision 59937)
+++ app/fcm_make/rose-app.conf (revision 60067)
@@ -42,4 +42,4 @@
stash_version=1A
timer_version=3A
-um_sources=branches/dev/lukeabraham/vn10.9_UKCA_Tutorial_Solns@46574
+um_sources=branches/dev/lukeabraham/vn10.9_UKCA_Tutorial_Solns@46619

These differences can be found in the file /home/ukca/Tutorial/vn10.9/worked_solutions/Task6.1/Task6.1_rose.patch on PUMA.

**vm:**

Index: app/fcm_make/rose-app.conf

--- app/fcm_make/rose-app.conf (revision 59931)
+++ app/fcm_make/rose-app.conf (revision 59986)
@@ -42,4 +42,4 @@
stash_version=1A
timer_version=3A
-um_sources=branches/dev/lukeabraham/vn10.9_UKCA_Tutorial_Solns@46574
+um_sources=branches/dev/lukeabraham/vn10.9_UKCA_Tutorial_Solns@46619

The specific UM changes made are:

Index: src/atmosphere/UKCA/ukca_chem_master.F90

--- src/atmosphere/UKCA/ukca_chem_master.F90 (revision 46574)
+++ src/atmosphere/UKCA/ukca_chem_master.F90 (revision 46619)
@@ -83,7 +83,7 @@
INTEGER, PARAMETER :: n_het_master = 10 ! number of heterogeneous reactions
INTEGER, PARAMETER :: n_dry_master = 57 ! number of dry deposition reactions
INTEGER, PARAMETER :: n_wet_master = 49 ! number of wet deposition reactions
-INTEGER, PARAMETER :: n_bimol_master= 401 ! number of bimolecular reactions
+INTEGER, PARAMETER :: n_bimol_master= 401 ! number of bimolecular reactions
INTEGER, PARAMETER :: n_ratj_master = 76 ! number of photolysis reactions
INTEGER, PARAMETER :: n_ratt_master = 49 ! number of termolecular reactions
@@ -2152,7 +2152,9 @@

The specific UM changes made are:

--- src/atmosphere/UKCA/ukca_chem_master.F90 (revision 46574)
+++ src/atmosphere/UKCA/ukca_chem_master.F90 (revision 46619)
@@ -83,7 +83,7 @@
INTEGER, PARAMETER :: n_het_master = 10 ! number of heterogeneous reactions
INTEGER, PARAMETER :: n_dry_master = 57 ! number of dry deposition reactions
INTEGER, PARAMETER :: n_wet_master = 49 ! number of wet deposition reactions
INTEGER, PARAMETER :: n_bimol_master= 401 ! number of bimolecular reactions
INTEGER, PARAMETER :: n_ratj_master = 76 ! number of photolysis reactions
INTEGER, PARAMETER :: n_ratt_master = 49 ! number of termolecular reactions
@@ -2152,7 +2152,9 @@

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_6

5/6
These differences can be found in the file /home/ukca/Tutorial/vn10.9/worked_solutions/Task6.1/Task6.1_code.patch on PUMA.

Sample output from this task can be found at /work/n02/n02/ukca/Tutorial/vn10.9/sample_output/Task6.1/atmosa.pa19810901_00 on ARCHER.

Checklist

☐ Add the new reaction into the correct reaction type array in ukca_chem_master.F90, incrementing the size of the array accordingly.

☐ If required, add special code to the asad_bimol.F90, asad_trimol.F90, ukca_hetero_mod.F90, or asad_hetero.F90 routines.

☐ For photolysis reactions, further work is required to calculate new cross sections. Code will also need to be added to ukca_strat_update.F90.

Tutorial 7

Written by Luke Abraham 2017


- This page was last modified on 3 January 2018, at 11:45.
UKCA Chemistry and Aerosol vn10.9 Tutorial 7

From UKCA

UKCA Chemistry and Aerosol Tutorials at vn10.9

Contents

- 1 What you will learn in this Tutorial
- 2 Task 7.1: adding new dry deposition values
- 3 Adding Dry Deposition
- 4 Chemistry Scheme Specification
- 5 2D Dry Deposition Scheme
- 6 Interactive Dry Deposition Scheme
  - 6.1 Changes to ukca_aerod.F90
  - 6.2 Changes to ukca_surfddr.F90
- 7 Solution to Task 7.1: adding new dry deposition values
- 8 Checklist

What you will learn in this Tutorial

In this tutorial you will learn how the two UKCA dry deposition schemes are implemented. You will then make changes to allow one of your new tracers to be dry-deposited.

Task 7.1: adding new dry deposition values

**TASK 7.1:** You should now add in the dry deposition of ALICE. This species deposits in a similar way to CO. The values for depvel_defs_strattrop are:

<table>
<thead>
<tr>
<th>Surface Type</th>
<th>Summer (day)</th>
<th>Summer (night)</th>
<th>Summer (24h ave)</th>
<th>Winter (day)</th>
<th>Winter (night)</th>
<th>Winter (24h ave)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Water</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Forest</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>Grass</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>Desert</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>Ice</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

i.e. the same as for CO.

**Hint**

You will need to make changes for both dry deposition schemes. The changes to ukca_surfddr.F90 can be made very easily by adding ALICE to the CO block in the CASE statement.

Adding Dry Deposition

UKCA uses two different dry-deposition schemes:

- A simple 2D parameterisation described by Giannakopoulos (1999)[1], Ganzeveld and Lelieveld (1995)[2], and Sander and Crutzen (1996)[3].
- A more detailed interactive parameterisation, based on the Wesely scheme (Wesely, 1989; Sanderson 2007)[4,5]

The default scheme is the interactive scheme, which is chosen by setting l_ukca_intdd to true in the UKCA panel.

**Note:** If you are using the interactive scheme and wish to add new values to it, you will also need to add values to the 2D scheme as well, otherwise you will get an error.

During this tutorial you will be tasked with adding the dry deposition of one of your new tracers.

References


Chemistry Scheme Specification

The default is to use the 2D scheme, although it is advisable to use the interactive scheme. Within the UKCA code, whether a species is dry deposited or not is controlled in the ukca_chem_master.F90 file. In the chch_defs_master array there are lines like

```fortran
10 DD: 7, WD: 4, chch_t1(10,'HONO2 ',1,'TR ',' ',1,1,0,TI+S+T+ST+R,0,0,107),&
1 10 DD: 8, WD: 5,
```

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_7
Changes to ukca_aerod.F90

When using this scheme, dry deposition occurs throughout the boundary layer, rather than just in the lowest model (i.e. surface) layer. The two routines ukca_aerod.F90 and ukca_surfddr.F90 contain species specific information, and it is these routines that need to be altered to add in values for a new species. Further details on this scheme can be found in the The UKCA UM documentation paper.

2D Dry Deposition Scheme

The deposition velocities for the 2D scheme are defined in the depvel_defs_master array, which is held in the ukca_chem_master.F90 module. This is a large derived type containing a length 30 array, usually formatted to be made up of size (6,5) blocks (for ease of reading), of the format

<table>
<thead>
<tr>
<th>N</th>
<th>SPECIES</th>
<th>QUALIFIER</th>
<th>DISQUALIFIER</th>
<th>VN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summer (day) velocity over water</td>
<td>Summer (night) velocity over water</td>
<td>Summer (24h ave.) velocity over water</td>
<td>Winter (day) velocity over water</td>
<td>Winter (night) velocity over water</td>
</tr>
<tr>
<td>Summer (day) velocity over forest</td>
<td>Summer (night) velocity over forest</td>
<td>Summer (24h ave.) velocity over forest</td>
<td>Winter (day) velocity over forest</td>
<td>Winter (night) velocity over forest</td>
</tr>
<tr>
<td>Summer (day) velocity over grass</td>
<td>Summer (night) velocity over grass</td>
<td>Summer (24h ave.) velocity over grass</td>
<td>Winter (day) velocity over grass</td>
<td>Winter (night) velocity over grass</td>
</tr>
<tr>
<td>Summer (day) velocity over desert</td>
<td>Summer (night) velocity over desert</td>
<td>Summer (24h ave.) velocity over desert</td>
<td>Winter (day) velocity over desert</td>
<td>Winter (night) velocity over desert</td>
</tr>
<tr>
<td>Summer (day) velocity over ice</td>
<td>Summer (night) velocity over ice</td>
<td>Summer (24h ave.) velocity over ice</td>
<td>Winter (day) velocity over ice</td>
<td>Winter (night) velocity over ice</td>
</tr>
</tbody>
</table>

and are in cm/s. The desert category is not used, and only the day and night values are considered in the calculation of the dry-deposition velocities. Examples of these values are

```
7
R and T are at older revision than S and ST. Make consistent
deptel_t(7,'O3 ', & ! (Ganzeveld & Lelieveld (1995)) note 1
   ! (modified to be the same as Guang's version)
   (0.05, 0.05, 0.05, 0.05, 0.05, 0.05) & ! 1.1
   0.85, 0.30, 0.65, 0.65, 0.25, 0.45) & ! 1.2
   0.65, 0.25, 0.45, 0.65, 0.25, 0.45) & ! 1.3
   0.18, 0.18, 0.18, 0.18, 0.18, 0.18 & ! 1.4
   0.05, 0.05, 0.05, 0.05, 0.05, 0.05) & ! 1.5
   T+S+T,0,0,107), &
deptel_t(7,'O3 ', & ! (Ganzeveld & Lelieveld (1995) - note 1)
   (0.07, 0.07, 0.07, 0.07, 0.07, 0.07) &
   1.00, 0.11, 0.56, 0.26, 0.11, 0.19) &
   1.00, 0.37, 0.69, 0.59, 0.46, 0.53) &
   0.26, 0.26, 0.26, 0.26, 0.26, 0.26) &
   0.05, 0.05, 0.05, 0.05, 0.05, 0.05) &
   T+S+R,0,0,107), &
deptel_t(7,'O3 ', & ! (Ganzeveld & Lelieveld (1995) - note 1)
   (0.07, 0.07, 0.07, 0.07, 0.07, 0.07) &
   1.00, 0.11, 0.56, 0.26, 0.11, 0.19) &
   1.00, 0.37, 0.69, 0.59, 0.46, 0.53) &
   0.26, 0.26, 0.26, 0.26, 0.26, 0.26) &
   0.07, 0.07, 0.07, 0.07, 0.07, 0.07) &
   S,0,0,107), &
9
No DD of NO in R scheme

deptel_t(8,'NO', & ! (inferred from NO2 - see Giannakopoulos (1998))
   (0.00, 0.00, 0.00, 0.00, 0.00, 0.00) & ! 2.1
   0.14, 0.01, 0.01, 0.01, 0.01, 0.01) & ! 2.2
   0.10, 0.01, 0.06, 0.01, 0.01, 0.01) & ! 2.3
   0.01, 0.01, 0.01, 0.01, 0.01, 0.01) & ! 2.4
   0.00, 0.00, 0.00, 0.00, 0.00, 0.00) & ! 2.5
   T+T+S+ST,0,0,107), &
```

Note: as you can see above, this definition makes use of the N, SCHEME, QUALIFIER, DISQUALIFIER, and VN format, and the settings for these are the same as in the adding new tracers tutorial, although here N should be incremented for each new deposition, where there might be the same species specified several times with changes to deposition velocities.

This scheme is controlled in ukca_ddeprt.F90. The deposition only occurs in the bottom (i.e. 'surface') layer.

Interactive Dry Deposition Scheme

Adding in new species to the interactive scheme is slightly more involved than for the 2D scheme. This scheme is controlled from the ukca_ddeptr.F90 routine which is called from ukca_chemistry_ctl.F90. The two routines ukca_aerod.F90 and ukca_surfddr.F90 contain species specific information, and it is these routines that need to be altered to add in values for a new species. Further details on this scheme can be found in the The UKCA UM documentation paper.

When using this scheme, dry deposition occurs throughout the boundary layer, rather than just in the lowest model (i.e. surface) layer.

Changes to ukca_aerod.F90

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_7
This routine calculates the aerodynamic and quasi-laminar surface resistances. The species dependant information that is needed is the diffusion coefficient, $d_0$ (in units of $\text{m}^2\text{s}^{-1}$). By default this is set to -1 if the species is not deposited. If it is deposited, and there are no values for this coefficient in the literature, it is suggested that $d_0_{\text{species}}$ is calculated as

$$d_{0,\text{species}} = \frac{d_{0,H_2O}}{M_{H_2O}/M_{\text{species}}}$$

Where $M_{H_2O}$ is the relative molecular mass of H$_2$O, and $M_{\text{species}}$ is the relative molecular mass of the species being deposited, and $d_{0,H_2O}$ is the diffusion coefficient for H$_2$O ($2.08\times10^{-5}$ $\text{m}^2\text{s}^{-1}$).

You should add in an appropriate value for the new species that you are depositing in the CASE statement in this routine. Examples of how this is already done are

```fortran
CASE ('\text{'03}\text{'02}\text{'03S}\text{'NO3}')
  d0(j) = 1.4e-5
CASE ('\text{'HONO}')
  d0(j) = 1.4e-5
CASE ('\text{'CO}')
  d0(j) = 1.4e-5
CASE ('\text{'NO2}\text{'NO3}\text{'O3}\text{'NO2}\text{'O3S}\text{'NO3}')
  d0(j) = d_{h2o} \times \text{SQRT}(m_{h2o}/m_{hono})
```

Note: If you have not yet defined a $M_{\text{species}}$ value for your new species, you will need to do this in `ukca_constants.F90`.

### Changes to ukca_surfddr.F90

The Wesely scheme considers 9, 13, 17, or 27 different surface types:

<table>
<thead>
<tr>
<th>9 Surface Types</th>
<th>13 Surface Types</th>
<th>17 Surface Types</th>
<th>27 Surface Types</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Plant Functional Types</strong></td>
<td><strong>Plant Functional Types</strong></td>
<td><strong>Plant Functional Types</strong></td>
<td><strong>Plant Functional Types</strong></td>
</tr>
<tr>
<td>16. Bare Soil</td>
<td>16. Bare Soil</td>
<td>16. Bare Soil</td>
<td>16. Bare Soil</td>
</tr>
<tr>
<td>17. Ice</td>
<td>17. Ice</td>
<td>17. Ice</td>
<td>17. Ice</td>
</tr>
</tbody>
</table>

The examples below are given for 9 surface types, but you will need to make changes for both options.

`ukca_surfddr.F90` sets the surface resistance (in $\text{cm}^2\text{s}^{-1}$) for each of the species dry-deposited (rsurf). If a species is not deposited onto a particular type of surface (but is deposited onto other types) then its resistance on this type can be set to a very large value ($r_{\text{null}}$). Often many species are assigned the same value. You will need to add in appropriate values for your species into the CASE statement within this routine.

Examples of how this is already done are

```fortran
CASE ('\text{'NO2}\text{'NO3}')
  rsurf(:,n)=(/225.,225.,400.,400.,600.,1200.,2600.,1200.,3500./)
CASE ('\text{'CO}')
  rsurf(:,n)=(/3700.,7300.,4550.,1960.,4550.0,\text{r_null},\text{r_null}, &
            4550.0,\text{r_null}/) ! Shrub+bare soil set to C3 grass (guess)
```

Remember to make the changes for ALL the case statements associated with the different numbers of surface types.

### Solution to Task 7.1: adding new dry deposition values

You were given the task

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_7
You should now add in the dry deposition of ALICE. This species deposits in a similar way to CO. The values for depvel_defs_strattrop are:

<table>
<thead>
<tr>
<th>Surface Type</th>
<th>Summer (day)</th>
<th>Summer (night)</th>
<th>Summer (24h ave)</th>
<th>Winter (day)</th>
<th>Winter (night)</th>
<th>Winter (24h ave)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Water</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Forest</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>Grass</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>Desert</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>Ice</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

i.e. the same as for CO.

You were given the hint:

- You will need to make changes for both dry deposition schemes. The changes to ukca_surfddr.f90 can be made very easily by adding ALICE to the CO block in the CASE statement.

For a working Rose suite that has completed this task, please see

- ARCHER: u-as292@60203
- vm: u-as297@60160

The specific Rose changes made are:

- ARCHER: https://code.metoffice.gov.uk/trac/roses-u/changeset/60203/a/s/2/9/2/trunk
- vm: https://code.metoffice.gov.uk/trac/roses-u/changeset/60160/a/s/2/9/7/trunk

The specific UM changes made are:

```
Index: app/fcm_make/rose-app.conf
===================================================================
--- app/fcm_make/rose-app.conf (revision 60067)
+++ app/fcm_make/rose-app.conf (revision 60203)
@@ -42,4 +42,4 @@
 stash_version=1A
 timer_version=3A
 um_rev=vn10.9
-um_sources=branches/dev/lukeabraham/vn10.9_UKCA_Tutorial_Solns@46619
+um_sources=branches/dev/lukeabraham/vn10.9_UKCA_Tutorial_Solns@46683

These differences can be found in the file /home/ukca/Tutorial/vn10.9/working_solutions/Task7.1/Task7.1_rose.patch on PUMA.
```

vm:

```
Index: app/fcm_make/rose-app.conf
===================================================================
--- app/fcm_make/rose-app.conf (revision 59986)
+++ app/fcm_make/rose-app.conf (revision 60160)
@@ -42,4 +42,4 @@
 stash_version=1A
 timer_version=3A
 um_rev=vn10.9
-um_sources=branches/dev/lukeabraham/vn10.9_UKCA_Tutorial_Solns@46619
+um_sources=branches/dev/lukeabraham/vn10.9_UKCA_Tutorial_Solns@46683

The specific UM changes made are:
```

```
Index: src/atmosphere/UKCA/ukca_chem_master.F90
===================================================================
--- src/atmosphere/UKCA/ukca_chem_master.F90 (revision 46619)
+++ src/atmosphere/UKCA/ukca_chem_master.F90 (revision 46683)
@@ -81,7 +81,7 @@
 ! define size of master chemistry
 INTEGER, PARAMETER :: n_chch_master = 150 ! number of known species
 INTEGER, PARAMETER :: n_het_master = 10 ! number of heterogeneous reactions
-INTEGER, PARAMETER :: n_dry_master = 57 ! number of dry deposition reactions
+INTEGER, PARAMETER :: n_dry_master = 58 ! number of dry deposition reactions
 INTEGER, PARAMETER :: n_wet_master = 49 ! number of wet deposition reactions
 INTEGER, PARAMETER :: n_bimol_master= 401 ! number of bimolecular reactions
 INTEGER, PARAMETER :: n_ratj_master = 76 ! number of photolysis reactions
@@ -377,7 +377,7 @@
 chch_t1(113,'MEMALD ','1','TR ',' ',0,0,0,0,0,0,107),&
 chch_t1(114,'GLY ','1','TR ',' ',0,0,0,0,0,107),&
 chch_t1(115,'OXYLENE ','1','TR ',' ',0,0,0,0,0,107),&
-chch_t1(116,'ALICE ','1','TR ',' ',0,0,0,0,0,107),&
+chch_t1(116,'ALICE ','1','TR ',' ',0,0,0,0,0,107),&
 chch_t1(117,'BOB ','1','TR ',' ',0,0,0,0,0,107) /
```

```
http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_7
4/6
! Heterogeneous chemistry

```
0.50, 0.50, 0.50, 0.50, 0.50, 0.50,&
0.50, 0.50, 0.50, 0.50, 0.50, 0.50,&
0.50, 0.50, 0.50, 0.50, 0.50, 0.50 /),
-TI,A,0,107) /)
+TI,A,0,107),
+! UKCA Tutorial Task7.1
+depvel_t(53,'ALICE     ',
+/0.00, 0.00, 0.00, 0.00, 0.00, 0.00,&
+ 0.03, 0.03, 0.03, 0.03, 0.03, 0.03,&
+ 0.03, 0.03, 0.03, 0.03, 0.03, 0.03,&
+ 0.00, 0.00, 0.00, 0.00, 0.00, 0.00 /),&
+ ST,0,0,107) /)
```

! determine which chemistry is to be used. Test here that only one scheme is
! selected.

Index: src/atmosphere/UKCA/ukca_surfddr.F90

```--- src/atmosphere/UKCA/ukca_surfddr.F90 (revision 46619)
+++ src/atmosphere/UKCA/ukca_surfddr.F90 (revision 46683)
@@ -433,7 +433,7 @@
  r_null,12500.0, 500.0,12500.0 /)
 CASE ('NH3       ')
   rsurf(:,n)=tenpointzero
 CASE ('CO        ')
+  CASE ('CO        ','ALICE     ')
    rsurf(:,n)=(/3700.0, 7300.0, 4550.0, 1960.0, 4550.0, &
               r_null, r_null, 4550.0, r_null /
               ! Shrub+bare soil set to C3 grass (guess)
@@ -517,7 +517,7 @@
  rsurf(1:6,n)=(/ 137.0,111.1,111.9,131.3,130.4,209.8 /)
 CASE ('CO        ')
+  CASE ('CO        ','ALICE     ')
    rsurf(1:6,n)=(/3700.0,3700.0,3700.0,7300.0,7300.0,4550.0 /)
               ! Shrub+bare soil set to C3 grass (guess)
CASE ('CH4      ')
@@ -594,7 +594,7 @@
  rsurf(7:16,n)=(/618.6,648.6,784.3,888.9,4000.0,1290.3,4000.0 /)
 CASE ('NH3       ')
   rsurf(7:16,n)=(/196.1,185.8,196.1,180.7,148.9,213.5,215.1 /
 CASE ('CO        ')
+  CASE ('CO        ','ALICE     ')
    rsurf(7:16,n)=(/1960.0,4550.0,4550.0,r_null,r_null,4550.0,r_null /
               ! Shrub+bare soil set to C3 grass (guess)
CASE ('CH4O     ')
@@ -648,7 +648,7 @@
  rsurf(7:17,n)=(/209.8,209.8,196.1,196.1,196.1,185.8,196.1,180.7, &
               148.9,213.5,215.1 /
 CASE ('CO        ')
+  CASE ('CO        ','ALICE     ')
    rsurf(7:17,n)=(/4550.0,4550.0,1960.0,1960.0,1960.0, &
               4550.0,4550.0,r_null,r_null,4550.0,r_null /
               ! Shrub+bare soil set to C3 grass (guess)
@@ -706,7 +706,7 @@
 CASE ('CO        ')
+  CASE ('CO        ','ALICE     ')
    rsurf(18:27,n)=(/r_null,r_null,r_null,r_null,r_null, &
                    r_null,r_null,r_null,r_null /
               ! Shrub+bare soil set to C3 grass (guess)
Index: src/atmosphere/UKCA/ukca_aerod.F90

```--- src/atmosphere/UKCA/ukca_aerod.F90 (revision 46619)
+++ src/atmosphere/UKCA/ukca_aerod.F90 (revision 46683)
@@ -231,6 +231,8 @@
  d0(j) = d_h2o * SQRT(m_h2o / m_meoh)
 CASE ('Monoterp  ')
  d0(j) = d_h2o * SQRT(m_h2o / m_monoterp)
+ CASE ('ALICE     ')
+  d0(j) = d_h2o * SQRT(m_h2o / m_ALICE)
END SELECT
```
These differences can be found in the file /home/ukca/Tutorial/vn10.9/worked_solutions/Task7.1/Task7.1_code.patch on PUMA.

Sample output from this task can be found at /work/n02/n02/ukca/Tutorial/vn10.9/sample_output/Task7.1/atmosa.pal9810901_00 on ARCHER.

Checklist

☐ Put a 1 in the D column of the chch_defs_master in ukca_chem_master.F90.
☐ Append deposition velocity values to the depvel_defs_master array, and increment n_dry_master.
☐ Put the correct M_species value in ukca_constants.F90.
☐ In ukca_aerod.F90, append the CASE statement with your new species to set a value for d0.
☐ In ukca_surfddr.F90, append all the CASE statements (for the various different numbers of surface types) with the values for your new species.

Tutorial 8

Written by Luke Abraham 2017

Retrieved from "http://www.ukca.ac.uk/wiki/index.php?title=UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_7&oldid=6460"

- This page was last modified on 11 December 2017, at 09:57.
What you will learn in this Tutorial

In this tutorial you will learn how the wet deposition of chemical species is handled in UKCA. You will then add-in the wet deposition of one of your new tracers.

Task 8.1: Add wet deposition of a species

Task 8.1: Add in wet deposition for BOB, using the following values:

<table>
<thead>
<tr>
<th>$k'(298)$</th>
<th>$-(\Delta H/R)$</th>
<th>$k'(298)$ for the 1st dissociation</th>
<th>$-(\Delta H/R)$ for the 1st dissociation</th>
<th>$k'(298)$ for the 2nd dissociation</th>
<th>$-(\Delta H/R)$ for the 2nd dissociation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0.21 \times 10^{+06}$</td>
<td>$0.87 \times 10^{+04}$</td>
<td>$0.2 \times 10^{+02}$</td>
<td>$0.0$</td>
<td>$0.0$</td>
<td>$0.0$</td>
</tr>
</tbody>
</table>

Adding Wet Deposition

The formulation used in UKCA is described in Giannakopoulos (1999)[1]. This scheme uses the following formula to calculate the effective Henry's Law coefficient

$$H_{eff} = k'(298) \exp \left( -\frac{\Delta H}{R} \left[ \left( \frac{1}{T} \right) - \left( \frac{1}{298} \right) \right] \right)$$

where $k'(298)$ is the rate constant at 298K.

During this tutorial you will be tasked with adding the wet deposition of one of your new tracers.

References


Turning on Wet Deposition for a Species

Chemistry Scheme Specification

Within the UKCA code, whether a species is wet deposited or not is controlled in the _ukca_chem_master.F90 module. In the _chch_defs_master_ array there are lines like

| 10 DD: 7,WD: 4, chch_t1(10,'HONO2     ',1,'TR    ','          ',1,1,0,TI+S+T+ST+R+0,0,107),& |
| 11 DD: 8,WD: 5, chch_t1(11,'H2O2      ',1,'TR    ','          ',1,1,0,TI+S+T+ST+OL+R+0,0,& |

Where the 1 in the 7th column turns on wet deposition of that species (being 0 otherwise). You will need to change the 0 to a 1 for the species that you wish to now wet deposit.

Setting Henry's Law values

In the _ukca_chem_master.F90 module the parameters required to calculate $H_{eff}$ are held in the _henry_defs_master_ array (of defined size n_wet_master), and has format

<table>
<thead>
<tr>
<th>N</th>
<th>SPECIES</th>
<th>$k'(298)$</th>
<th>$-(\Delta H/R)$</th>
<th>$k'(298)$ for the 1st dissociation</th>
<th>$-(\Delta H/R)$ for the 1st dissociation</th>
<th>$k'(298)$ for the 2nd dissociation</th>
<th>$-(\Delta H/R)$ for the 2nd dissociation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCHEME</td>
<td>QUALIFIER</td>
<td>DISQUALIFIER</td>
<td>VN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Columns 3 and 4 are used if the species dissociates in the aqueous phase. In this case, $H_{aq}$ is further multiplied by a factor of

$$1 + \frac{k(aq)}{H^+}$$

where

$$k(aq) = k(298) \exp \left( -\frac{\Delta H}{R} \left[ \left( \frac{1}{T} \right) - \left( \frac{1}{298} \right) \right] \right)$$

and column 3 contains the values of $k(298)$ and column 4 contains the value of $-\Delta H / R$. Similarly, if the species dissociates a second time then a further factor of $1 + k(aq)/H^+$ is applied, where this value of $k(aq)$ is calculated from the values of $k(298)$ and $-\Delta H / R$ in columns 5 and 6.

**Note:** As with the 2D dry deposition values in `depvel_defs_master`, the order of `henry_defs_master` also assumes that the values are in the same order as the species (that wet deposit) in the `chch_defs_master` array.

Examples for this array are:

```plaintext
4  wetdep(4,'HONO2 ',&
    \(0.21e+06,0.87e+04,0.20e+02,0.00e+00,0.00e+00,0.00e+00/)\),TI+T+ST+R,0,0,107),&
4  wetdep(4,'HONO2 ',&
    \(0.21e+06,0.87e+04,0.157e+02,0.00e+00,0.00e+00,0.00e+00/)\),S,0,0,107),&
5  wetdep(5,'H2O2 ',&
    \(0.83e+05,0.74e+04,0.24e-11,-0.373e+04,0.e+00,0.e+00/)\),TI+T+ST+OL+R,0,0,107),&
5  wetdep(5,'H2O2 ',&
    \(0.83e+05,0.74e+04,0.22e-11,-0.373e+04,0.00e+00,0.00e+00/)\),S,0,0,107),&
```

**Solution to Task 8.1: Add wet deposition of a species**

You were given the task

- **Add in wet deposition for BOB**, using the following values:

<table>
<thead>
<tr>
<th>$k(298)$</th>
<th>$-\frac{\Delta H}{R}$ for the 1st dissociation</th>
<th>$k(298)$ for the 1st dissociation</th>
<th>$-\frac{\Delta H}{R}$ for the 2nd dissociation</th>
<th>$k(298)$ for the 2nd dissociation</th>
<th>$-\frac{\Delta H}{R}$ for the 2nd dissociation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0.21 \times 10^{+06}$</td>
<td>$0.87 \times 10^{+04}$</td>
<td>$0.2 \times 10^{+02}$</td>
<td>$0.0$</td>
<td>$0.0$</td>
<td>$0.0$</td>
</tr>
</tbody>
</table>

For a working Rose suite that has completed this task, please see

- **ARCHER**: u-as292@60205
- **vm**: u-as297@60206

The specific Rose changes made are:

- **ARCHER**: https://code.metoffice.gov.uk/trac/roses-u/changeset/60205/a/s/2/9/2/trunk
- **vm**: https://code.metoffice.gov.uk/trac/roses-u/changeset/60206/a/s/2/9/7/trunk

The specific Rose changes made are:

**ARCHER**:

```plaintext
Index: app/fcm_make/rose-app.conf
===================================================================
--- app/fcm_make/rose-app.conf (revision 60203)
+++ app/fcm_make/rose-app.conf (revision 60205)
@@ -42,4 +42,4 @@
 stash_version=1A
 timer_version=3A
 um_rev=vn10.9
-um_sources=branches/dev/lukeabraham/vn10.9_UKCA_Tutorial_Solns#46683
+um_sources=branches/dev/lukeabraham/vn10.9_UKCA_Tutorial_Solns#46696
```

These differences can be found in the file `/home/ukca/Tutorial/vn10.9/worked_solutions/Task8.1/Task8.1_rose.patch` on PUMA.

**vm**:

```plaintext
Index: app/fcm_make/rose-app.conf
===================================================================
--- app/fcm_make/rose-app.conf (revision 60160)
+++ app/fcm_make/rose-app.conf (revision 60206)
@@ -42,4 +42,4 @@
 stash_version=1A
 timer_version=3A
```

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_8
The specific UM changes made are:

--- src/atmosphere/UKCA/ukca_chem_master.F90
+++ src/atmosphere/UKCA/ukca_chem_master.F90  (revision 46683)
@@ -82,7 +82,7 @@
     INTEGER, PARAMETER :: n_chch_master = 150  ! number of known species
     INTEGER, PARAMETER :: n_het_master  =  10  ! number of heterogeneous reactions
     INTEGER, PARAMETER :: n_dry_master  =  58  ! number of dry deposition reactions

-    INTEGER, PARAMETER :: n_wet_master  =  49  ! number of wet deposition reactions
+    INTEGER, PARAMETER :: n_wet_master  =  50  ! number of wet deposition reactions
     INTEGER, PARAMETER :: n_bimol_master= 401  ! number of bimolecular reactions
     INTEGER, PARAMETER :: n_ratj_master =  76  ! number of photolysis reactions
     INTEGER, PARAMETER :: n_ratt_master =  49  ! number of termolecular reactions
@@ -378,7 +378,7 @@
     chch_t1(114,'GLY       ',1,'TR        ','          ',0,1,0,R,0,0,107),&
     chch_t1(115,'oXYLENE   ',1,'TR        ','          ',0,0,1,R,0,0,107),&
-    chch_t1(117,'BOB       ',1,'TR        ','          ',0,0,0,ST,0,0,107)/)
+    chch_t1(117,'BOB       ',1,'TR        ','          ',0,1,0,ST,0,0,107)/)

! Heterogeneous chemistry
! Columns take the following meanings:
@@ -1039,7 +1039,10 @@
     wetdep(38,'s-BuOOH   ',
-     (/(0.34e+03,0.00e+00,0.00e+00,0.00e+00,0.00e+00,0.00e+00/),R,0,0,107), &
     wetdep(39,'GLY     ',
+    (/(0.36e+06,0.00e+00,0.00e+00,0.00e+00,0.00e+00,0.00e+00/),R,0,0,107) /)
+    UKCA Tutorial: wet deposition of BOB
+    wetdep(40,'BOB    ',
+    (/(0.21e+06,0.87e+04,0.20e+02,0.00e+00,0.00e+00,0.00e+00/),ST,0,0,107) /)

! Bimolecular reactions are too many to define here in one statement.
TYPE(ratb_t1) :: ratb_defs_master(1:n_bimol_master)

These differences can be found in the file /home/ukca/Tutorial/vn10.9/worked_solutions/Task8.1/Task8.1_code.patch on PUMA.

Sample output from this task can be found at /work/n02/n02/ukca/Tutorial/vn10.9/sample_output/Task8.1/atmosa.pali9810901_00 on ARCHER.

Checklist

☐ Put a 1 in the W column of the chch_defs_master in ukca_chem_master.F90.
☐ Append the Henry’s Law parameter values in the depvel_defs_master array, and increment n_wet_master.

Tutorial 9

Written by Luke Abraham 2017


- This page was last modified on 11 December 2017, at 09:58.
UKCA Chemistry and Aerosol vn10.9 Tutorial 9

From UKCA
UKCA Chemistry and Aerosol Tutorials at vn10.9

Contents

1 What you will learn in this Tutorial
2 Task 9.1: Output new diagnostics
3 Adding New UKCA Diagnostics
4 Flux Definitions in asad_flux_dat.F90

4.1 Diagnostic Type
4.2 STASH Code
4.3 Diagnostic Specification
4.4 Mask
4.5 Reaction number
4.6 Number of Species
4.7 Species
4.8 Reactants and Products
4.9 Summing Diagnostics
4.10 Changes to asad_flux_dat.F90
4.11 Changes to asad_chem_flux_diags.F90 (and other UKCA routines)

5 STASHmaster file

5.1 Rose Changes

6 Solution to Task 9.1: Output new diagnostics

7 Checklist

What you will learn in this Tutorial

In this tutorial you will learn about the UKCA diagnostics package and the different diagnostics that you can output using it. You will also learn how to add new diagnostics from the new reactions and deposition that you have added.

Task 9.1: Output new diagnostics

TASK 9.1: Output diagnostics of the reaction $\text{ALICE} + \text{OH} \rightarrow \text{BOB} + \text{SEC} + \text{ORG}$ to STASH code 50134, the dry deposition of ALICE to STASH code 50135, and the wet deposition of BOB to 50136. They should be outputted as a 3-hour mean to the pa/UPA stream.

Adding New UKCA Diagnostics

If you are using one of the chemistry schemes that uses ASAD (e.g. CheT/TropI sop, CheS/Strat, CheST/StratTrop) then you can make use of the ASAD Reaction Flux Diagnostics module (held in asad_chem_flux_diags.F90). These allow you to straightforwardly output new reaction and deposition fluxes.

To output new diagnostics you will first need to define them in the asad_flux_dat.F90 module, and then create new STASHmaster file specifications for them.

During this tutorial you will be tasked with outputting the reaction and deposition fluxes that you have added in to your branch.

Flux Definitions in asad_flux_dat.F90

Within the asad_flux_dat.F90 module the diagnostics are defined in blocks with the format

```fortran
asad_flux_defn('Diagnostic type',STASH code,'Diagnostic specification',Mask,Reaction number,Number of species,
               '/Species/Reactant 1','Reactant 2'/),
               '/Product 1','Product 2','Product 3','Product 4'/),
```

Which have the following meaning:

**Diagnostic Type**

This is a **three character string** which defines what type of diagnostic is being requested. This can take the values

- **RXN** to output the flux through a reaction (in moles/gridcell/s)
- **DEP** to output the deposition flux of a species (in moles/gridcell/s)
- **NET** to output the net chemical tendency of a species (in moles/gridcell/s)
- **STE** to output the net dynamical tendency of a species (in moles/gridcell/s)
- **MAS** to output the mass of the atmosphere (in kg/gridcell)
- **PSC** to output polar stratospheric cloud diagnostics (1 when the gridcell contains a PSC, 0 otherwise - monthly mean field will be a fraction in range 0 → 1)
- **TPM** to output the tropospheric mask (1 for troposphere, 0 otherwise - monthly mean field will be a fraction in range 0 → 1)
- **OUT** to output a tracer in mmr. Only really useful if the field is masked to give the tropospheric concentration only (see the discussion of the Mask option)

**STASH Code**

This is a **5 digit integer** defining the STASH code that the diagnostic will be outputted to (e.g. 50001). Currently this must be in section 50.

**Diagnostic Specification**

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_9
This is a one character string which is needed to further define what diagnostic is required. If it isn't needed then it should just be set to X or left blank.

- RXN
  - B to output the flux through a bimolecular reaction
  - T to output the flux through a termolecular reaction
  - H to output the flux through a heterogeneous reaction
  - J to output the flux through a photolysis reaction

- DEP
  - D to output the dry deposition flux
  - W to output the wet deposition flux

- PSC
  - 1 to output the fraction of Type 1 PSCs
  - 2 to output the fraction of Type 2 PSCs

Mask

This is a logical which defines whether only the tropospheric values of the diagnostic are outputted (.TRUE.) or not (.FALSE.). It is calculated every timestep.

For the STE diagnostic this is required if you wish to output the diagnosed stratosphere-troposphere exchange of a species. For the OUT diagnostic this can be used to output only the tropospheric concentration of a tracer. This is also used in the calculation of the of the TPM diagnostic.

Reaction number

This is an integer, and should only be used in the special case of there being two (or more) reactions with the exactly the same reactants and products, but with different rate coefficients. In this case the first reaction in the list would be given number 1 and the second 2 etc. If this is not needed then it should be set to 0 (which will be usual for most reactions).

Number of Species

This is an integer, and should give the total number of species, so this will be 1 for diagnostics such as DEP, STE, NET etc., which only consider a single species, and the total number of reactants and products for diagnostics RXN and RTE.

Species

This is a 10-character string giving the exact name of the species that the diagnostic should be considered for (including capitalisation). This is only used for the DEP, NET, STE, and OUT. For the RXN and RTE diagnostics the full list of reactants and products should be given (see below). For the MAS, PSC, and TPM diagnostics this isn't needed and could either be set to XXX or left blank. If it is needed the other reactant/product slots should be left blank.

Reactants and Products

These are 10-character strings, and should be as the reaction is defined in the ukca_chem_master.F90 module.

Summing Diagnostics

If you define more than one diagnostic to be output to the same STASH code, then the diagnostic routines will sum these diagnostics together. This can be useful (e.g., if you wanted to output the sum of all NO+RO2 reactions to one STASH item), but can be problematic if you accidentally output two fields to the same STASH code, as this will give strange results!

Changes to asad_flux_dat.F90

After you have defined your new diagnostics at the top of this module, you will need to make sure that they have been added correctly to the asad_chemical_fluxes array, which is defined in the asad_load_default_fluxes subroutine held in the asad_flux_dat.F90 module. You will need to increment the size of this array, given by the n_chemical_fluxes parameter.

Changes to asad_chem_flux_diags.F90 (and other UKCA routines)

You will need to edit asad_chem_flux_diags.F90 module if you want to output a new type of diagnostic. This can be quite involved, but you can look at existing routines to see how things are done. You will also need to add code into the main UKCA routines the pass the data through, e.g.

```fortran
! 3D flux diagnostics
IF (L_asad_use_chem_diags .AND.                   &
   ((L_asad_use_flux_rxns .OR. L_asad_use_rxn_rates) .OR. &
   (L_asad_use_wetdep .OR. L_asad_use_drydep)) )        &
   CALL asad_chemical_diagnostics(row_length,rows,      &
    model_levels,ix,jy,k,secs_per_step,volume,ierr)
```

STASHmaster file

While the diagnostics are defined in asad_flux_dat.F90 they are turned on by requesting the item through STASH. To do this you will need to edit the STASHmaster_A file in your branch.

Most UKCA diagnostics are 3D, although some, such as emissions (which are outputted in a different way), are 2D. You should take care with the STASH settings in STASHmaster_A between these types, as there some differences that will need to be considered.

Rose Changes

If you have not done so, you will also need to make sure that you use your new STASHmaster_A file in Rose, as is explained in detail in Tutorial 4.

After you have made your STASmaster_A file changes, you will need to add these diagnostics into STASH, as per Tutorial 3. Remember to run the TidyStashTransform macro. As you will have included your branch's STASHmaster_A file using @HEAD, you won't need to make any further changes to Rose, but you will need to make sure that you have committed your UM branch prior to running. Because the file is taken from £cm:run_xm_br you will need to make sure that the revision has synced to the PUMA mirror (i.e. by waiting a few minutes).
Solution to Task 9.1: Output new diagnostics

ALICE + OH → BOB + Sec_Org (s50i230) reaction flux (mol/s) at the surface (20m)

ALICE dry deposition flux (s50i231) (mol/s) at the surface (20m)

BOB wet deposition flux (s50i232) (mol/s) at the surface (20m)

You were given the task

- Output diagnostics of the reaction **ALICE + OH → BOB + SEC_ORG** to STASH code **50134**, the dry deposition of **ALICE** to STASH code **50135**, and the wet deposition of **BOB** to **50136**. They should be outputted as a 3-hour mean to the pa/UPA stream.

For a working Rose suite that has completed this task, please see

- ARCHER: u-ss292@60289
- vm: u-ss297@60286

The specific Rose changes made are:

- **ARCHER**: https://code.metoffice.gov.uk/trac/roses-u/changeset/60289/a/s/2/9/2/trunk
- **vm**: https://code.metoffice.gov.uk/trac/roses-u/changeset/60286/a/s/2/9/7/trunk

The specific Rose changes made are:

ARCHER:
These differences can be found in the file `/home/ukca/Tutorial/vn10.9/worked_solutions/Task9.1/Task9.1_rose.patch` on PUMA.

vm:
The specific UM changes made are:

```fortran
! Number of chemical fluxes defined below
INTEGER, PARAMETER :: n_chemical_fluxes = 310
+INTEGER, PARAMETER :: n_chemical_fluxes = 313

TYPE(asad_flux_defn), ALLOCATABLE, SAVE :: asad_chemical_fluxes(:)

++TYPE(asad_flux_defn), PARAMETER, PUBLIC ::
    ukca_tutorial_fluxes(3) = (/ &
     asad_flux_defn('RXN',50134,'B',.FALSE.,0,4, &
     /'ALICE     ','OH        '/), &
     asad_flux_defn('DEP',50135,'D',.FALSE.,0,1, &
     /'ALICE     ','Sec_Org   ','          ','          '</)), &
     asad_flux_defn('DEP',50136,'W',.FALSE.,0,1, &
     /'BOB       ','          '/), &
     asad_flux_defn('DEP',50136,'W',.FALSE.,0,1, &
     /'BOB       ','          '/), &
     asad_flux_defn('DEP',50136,'W',.FALSE.,0,1, &
     /'BOB       ','          '/), &
     asad_flux_defn('DEP',50136,'W',.FALSE.,0,1, &
     /'BOB       ','          '/)) &

++TYPE(asad_flux_defn), PUBLIC :: asad_aerosol_chem(16)

PUBLIC :: asad_load_default_fluxes

++ -1367,7 +1368,8 @@
    asad_ro2ro2_reacn, & ! 8 263
    asad_ch4_oxidn, & ! 6 269
    asad_old_prod, & ! 3 272
    asad_h2o_budget & ! 30 307
    asad_h2o_budget, & ! 30 307
    ukca_tutorial_fluxes & ! 3

IF (printstatus > PrStatus_Normal) THEN

```

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_9
**moles/s**

```plaintext
[stashmaster:code(50136)]
description=WET DEP FLUX: BOB (3D)
help=Wet Deposition flux of BOB (3D)
```  

```plaintext
[stashmaster:code(50140)]
description=DMS + OH => SO2 + MeOO + HCHO
help=Chemical reaction flux for DMS + OH => SO2 + MeOO + HCHO
```

```plaintext
Index: rose-meta/um-atmos/HEAD/etc/stash/STASHmaster/STASHmaster_A
--- rose-meta/um-atmos/HEAD/etc/stash/STASHmaster/STASHmaster_A (revision 46696)
+++ rose-meta/um-atmos/HEAD/etc/stash/STASHmaster/STASHmaster_A (revision 46718)
@@ -23558,6 +23558,24 @@
 5|    0 | 1871 |    0 |    8 |    0 |    0 |    0 |    0 |    0 |
#
+1|    1 |   50 |  135 |DRY DEP FLUX: ALICE (3D)            |
+2|    0 |    0 |   17 |    1 |    2 |    10 |    11 |    0 |    0 |    0 |    0 |
+3| 000000000000000000000000000000 | 00000000000000000001 |    3 |
+5| 0 | 1871 |    0 |    8 |    0 |    0 |    0 |    0 |    0 |
#
+1|    1 |   50 |  136 |WET DEP FLUX: BOB (3D)              |
+2|    0 |    0 |   17 |    1 |    2 |    10 |    11 |    0 |    0 |    0 |    0 |
+3| 000000000000000000000000000000 | 00000000000000000001 |    3 |
+5| 0 | 1871 |    0 |    8 |    0 |    0 |    0 |    0 |    0 |
#
+1|    1 |   52 |  134 |RXN FLUX: ALICE+OH->BOB+Sec_Org    |
+2|    0 |    0 |   17 |    1 |    2 |    10 |    11 |    0 |    0 |    0 |    0 |
+3| 000000000000000000000000000000 | 00000000000000000001 |    3 |
+5| 0 | 1871 |    0 |    8 |    0 |    0 |    0 |    0 |    0 |
#
+1|    1 |   52 |  136 |WET DEP FLUX: BOB (3D)              |
+2|    0 |    0 |   17 |    1 |    2 |    10 |    11 |    0 |    0 |    0 |    0 |
+3| 000000000000000000000000000000 | 00000000000000000001 |    3 |
+5| 0 | 1871 |    0 |    8 |    0 |    0 |    0 |    0 |    0 |
#
+1|    1 |   52 |  135 |DMS + OH => SO2 + MeOO + HCHO       |
+2|    0 |    0 |   17 |    1 |    2 |    10 |    11 |    0 |    0 |    0 |    0 |
+3| 000000000000000000000000000000 | 00000000000000000001 |    3 |
+5| 0 | 1871 |    0 |    8 |    0 |    0 |    0 |    0 |    0 |
#
+1|    1 |   52 |  134 |RXN FLUX: ALICE+OH->BOB+Sec_Org PLEV|
+2|    0 |    0 |   17 |    1 |    2 |    10 |    11 |    0 |    0 |    0 |    0 |
+3| 000000000000000000000000000000 | 00000000000000000001 |    3 |
+5| 0 | 1871 |    0 |    8 |    0 |    0 |    0 |    0 |    0 |
#
+1|    1 |   52 |  136 |WET DEP FLUX: BOB ON PRESS LEVELS   |
+2|    0 |    0 |   17 |    1 |    2 |    10 |    11 |    0 |    0 |    0 |    0 |
+3| 000000000000000000000000000000 | 00000000000000000001 |    3 |
+5| 0 | 1871 |    0 |    8 |    0 |    0 |    0 |    0 |    0 |
#
+1|    1 |   52 |  135 |DMS + OH => SO2 + MeOO + HCHO PRESS |
+2|    0 |    0 |   17 |    1 |    2 |    10 |    11 |    0 |    0 |    0 |    0 |
+3| 000000000000000000000000000000 | 00000000000000000001 |    3 |
+5| 0 | 1871 |    0 |    8 |    0 |    0 |    0 |    0 |    0 |
#
These differences can be found in the file /home/ukca/Tutorial/vn10.9/worked_solutions/Task9.1/Task9.1_code.patch on PUMA.

If you open the .pa file in Xconv, you should see the following fields:

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>96</td>
<td>72</td>
<td>38</td>
<td>1</td>
<td>molfluxd: Stash code = 50134</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>96</td>
<td>72</td>
<td>38</td>
<td>1</td>
<td>molfluxd: Stash code = 50135</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>96</td>
<td>72</td>
<td>38</td>
<td>1</td>
<td>molfluxd: Stash code = 50136</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sample output from this task can be found at /work/n02/n02/ukca/Tutorial/vn10.9/sample_output/Task9.1/atmosa.pa19810901_00 on ARCHER.

Checklist

☐ Make the required changes to branch's STASHmaster_A file, to add the new diagnostics. Make a note of the STASH items chosen, and also add the pressure-level diagnostics.
☐ Add help text for your diagnostics in branch's STASH/master-meta.conf file.
☐ Using a text editor, open the app/um/rose-app.conf file from your roses/[SUITE-ID] directory, and add the line STASHMASTER=STASHmaster in the [env] block, then save and close the file.
☐ Using a text editor, open the rose-suite.conf file from your roses/[SUITE-ID] directory, and add the following lines to the top of the file, before saving and closing it:

```
[file:app/um/file/STASHmaster]
source=fcm:um.xm_br/dev/[your MOSRS userid]/vnX.Y_your_branch_name/rose-meta/um-atmos/HEAD/etc/stash/STASHmaster
```

☐ Edit your suite using rose edit -M /path/to/your/branch/working/copy.
☐ Point the metadata in your suite to um-atmos/HEAD.
☐ Include your branch in your suite at: fcm_make → env → Sources.
☐ In asad_flux_dat.F90, make a new array of type asad_flux_defn and populate it with your new diagnostic specification(s), referencing the same item numbers as in your STASHmaster_A file.
☐ Append this new array at the end of asad_chemical_fluxes, and increment n_chemical_fluxes by the number of new diagnostics.
☐ If required, add new code for new diagnostics into asad_chem_flux_diags.F90 and other UKCA routines as necessary.
☐ Output your diagnostics in STASH at: um → namelist → Model Input and Output → STASH Requests and Profiles → STASH Requests.
☐ Run the TidyStashTransform transform macro.
☐ Save your suite.
☐ In the roses/[SUITE-ID] directory, run fcm commit to commit your changes to the repository.
☐ Run your suite.

Tutorial 10

Written by Luke Abraham 2017


- This page was last modified on 3 January 2018, at 11:52.
UKCA Chemistry and Aerosol vn10.9 Tutorial 10

From UKCA

UKCA Chemistry and Aerosol Tutorials at vn10.9

Contents

- 1 What you will learn in this Tutorial
  - 1.1 Python
- 2 Task 10.1: Output aerosol diagnostics
- 3 Radiation Timesteps
- 4 Making new STASH profiles
  - 4.1 Time profiles
  - 4.1.1 Example: Output on Radiation Timesteps
  - 4.2 Domain profiles
  - 4.2.1 Example: Output AOT pseudo levels on all model theta levels
  - 4.3 Usage profiles
  - 4.4 Using your new profiles
- 5 Solution to Task 10.1
- 6 Task 10.2: Calculate aerosol optical depth
  - 6.1 Python script
- 7 Solution to Task 10.2
- 8 Task 10.3: Calculate the single-scattering albedo
  - 8.1 Python script
- 9 Solution to Task 10.3
- 10 Task 10.4: Calculate the top of the atmosphere net downward radiative flux
  - 10.1 Python script
- 11 Solution to Task 10.4
- 12 Task 10.5: Calculate aerosol optical depth from the 3D aerosol extinction
  - 12.1 Python script
- 13 Solution to Task 10.5
- 14 Task 10.6: Calculate the difference in aerosol impacts when Sec_Org is no longer formed from ALICE
- 15 Code changes required to modify a reaction
  - 15.1 ukca_chem_master.F90
  - 15.3 asad_flux_dat.F90
- 16 Solution to Task 10.6
- 17 Checklist

What you will learn in this Tutorial

In this tutorial you will learn how to output and process aerosol diagnostics from UKCA.

Python

In this Tutorial you will make extensive use of Python to process the UM output. Instructions as to how to use this are in Tutorial 5.

Example python scripts are provided for each Task, and you should take the time to read through these and understand what they are doing. None of these scripts plot any of the output, so as an extension to these tasks you could try extending the scripts to plot the output from python directly, rather than visualising via Xconv.

Further information (and examples) on how Iris plots can be found here:

- http://scitools.org.uk/iris/docs/latest/userguide/plotting_a_cube.html

Task 10.1: Output aerosol diagnostics

**TASK 10.1:** Output the following aerosol and radiation diagnostics to the UPA output stream. You should make a new time profile (called TRAD) to only output these on radiation timesteps. You will also need to make a new domain profile (called D3DAR) for s02i530 and s02i540, to output these on both model levels and pseudo levels.

<table>
<thead>
<tr>
<th>Hint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remember to output your aerosol diagnostics on the AOT pseudo levels using the DIAGAOT profile.</td>
</tr>
</tbody>
</table>
Raphson chemical solver is first called on the first hour, and then every hour after that. For the UKCA training suite this would mean that we need to start at timestep 1, as radiation is called on the first timestep, and then called every third model timestep after that. In contrast, the UKCA Newton-Raphson chemical solver is first called on the first hour, and then called every third model timestep after that. In the UKCA training suite, both of these are set to 16, i.e. a call every 90 minutes. This value is very configuration dependent - sometimes radiation is called every hour (24), sometimes every 3 hours (8) etc. You should check the settings in your suite if you need to consider any of the radiation diagnostics.

Making new STASH profiles

While we have covered outputting and creating new diagnostics in Tutorial 3, Tutorial 4, Tutorial 5, and Tutorial 9, you were only making use of existing profiles. Here you will learn how to make new time, domain, and usage profiles.

Time profiles

To make a new time profile, go to: um → namelist → Model input and Output → STASH Requests and Profiles → Time Profiles. This contains a list of existing time profiles, with names like t3hmn_039ecafe etc.

To make a new time profile, you should:

1. Right-click anywhere on the list and click the blue plus symbol (+) named new section
   This will make a new blank line (usually labelled 1, the next 2 etc.) with a red X next to it.

2. Right-click this new line and click View namelist.

3. Fill-in the values as required for what you want to do.

4. When you have finished, you will need to run the When you have finished, you will need to run the stashindices.TidyStashTransform macro by going to the Metadata → um drop-down menu. This will rename the new profile into the correct format.

Example: Output on Radiation Timesteps

You will have discovered that there are 16 radiation timesteps per day, but to work out how to output diagnostics you need to know many model timesteps per day there are. This can be found at: um → namelist → Top Level Model Control → Model Domain and Timestep. The number is given by the variable steps_per_periodim, which is defined as the number of steps per period given in seconds by secs_per_periodim. For one day, this value is 86400.

In the UKCA training suite, secs_per_periodim=86400, and steps_per_periodim=48. This means that there are 48 timesteps per day, i.e. each timestep is 30 minutes.

We can combine this with the number of radiation timesteps per day (16), to calculate that radiation is called every third model timestep (48/16 = 3) in this configuration. We can now fill-in the values in the new profile accordingly, e.g.

- ithp: No time processing
- tim_name: e.g. TRAD
- until: Timesteps (1)
- iopt: Regular intervals
- istr: 1 (i.e. start on first timestep)
- iend: -1 (i.e. never stop outputting)
- ifre: 3 (i.e. output every third timestep)

We need to start at timestep 1, as radiation is called on the first timestep, and then called every third model timestep after that. In contrast, the UKCA Newton-Raphson chemical solver is first called on the first hour, and then every hour after that. For the UKCA training suite this would mean that istr=2 and ifre=2.

---

<table>
<thead>
<tr>
<th>STASH Section</th>
<th>STASH Item</th>
<th>STASH Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>207</td>
<td>INCOMING SW RAD FLUX (TOA): ALL TSS</td>
</tr>
<tr>
<td>1</td>
<td>208</td>
<td>OUTGOING SW RAD FLUX (TOA)</td>
</tr>
<tr>
<td>2</td>
<td>205</td>
<td>OUTGOING LW RAD FLUX (TOA)</td>
</tr>
<tr>
<td>2</td>
<td>285</td>
<td>MINERAL DUST OPTICAL DEPTH IN RADN.</td>
</tr>
<tr>
<td>2</td>
<td>300</td>
<td>AITKEN MODE (SOLUBLE) OPTICAL DEPTH</td>
</tr>
<tr>
<td>2</td>
<td>301</td>
<td>ACCUM MODE (SOLUBLE) OPTICAL DEPTH</td>
</tr>
<tr>
<td>2</td>
<td>302</td>
<td>COARSE MODE (SOLUBLE) OPTICAL DEPTH</td>
</tr>
<tr>
<td>2</td>
<td>303</td>
<td>AITKEN MODE (INSOL) OPTICAL DEPTH</td>
</tr>
<tr>
<td>2</td>
<td>304</td>
<td>ACCUM MODE (INSOL) OPTICAL DEPTH</td>
</tr>
<tr>
<td>2</td>
<td>305</td>
<td>COARSE MODE (INSOL) OPTICAL DEPTH</td>
</tr>
<tr>
<td>2</td>
<td>585</td>
<td>MINERAL DUST ABS. OPTICAL DEPTH</td>
</tr>
<tr>
<td>2</td>
<td>240</td>
<td>AITKEN (SOLUBLE) ABS OPTICAL DEPTH</td>
</tr>
<tr>
<td>2</td>
<td>241</td>
<td>ACCUM (SOLUBLE) ABS OPTICAL DEPTH</td>
</tr>
<tr>
<td>2</td>
<td>242</td>
<td>COARSE (SOLUBLE) ABS OPTICAL DEPTH</td>
</tr>
<tr>
<td>2</td>
<td>243</td>
<td>AITKEN (INSOL) ABS OPTICAL DEPTH</td>
</tr>
<tr>
<td>2</td>
<td>244</td>
<td>ACCUM (INSOL) ABS OPTICAL DEPTH</td>
</tr>
<tr>
<td>2</td>
<td>245</td>
<td>COARSE (INSOL) ABS OPTICAL DEPTH</td>
</tr>
<tr>
<td>2</td>
<td>530</td>
<td>UKCA 3D AEROSOL EXTINCTION</td>
</tr>
<tr>
<td>2</td>
<td>540</td>
<td>CLASSIC 3D AEROSOL EXTINCTION</td>
</tr>
</tbody>
</table>
Domain profiles

To make a new time profile, go to: $um \rightarrow namelist \rightarrow Model Input and Output \rightarrow STASH Requests and Profiles \rightarrow Domain Profiles$. This contains a list of existing domain profiles, with names like $dallch_1e2e730d$. etc.

To make a new domain profile, you should:

1. Right-click anywhere on the list and click the blue plus symbol (+) named new section
   This will make a new blank line (usually labelled 1, the next 2 etc.) with a red X next to it.

2. Right-click this new line and click View namelist.

3. Fill-in the values as required for what you want to do.

4. When you have finished, you will need to run the When you have finished, you will need to run the $stashindices.TidyStampTransform$ macro by going to the Metadata $\rightarrow um$ drop-down menu. This will rename the new profile into the correct format.

Example: Output AOT pseudo levels on all model theta levels

Aerosol diagnostics are defined on a series of 6 pseudo levels, corresponding to 0.38, 0.44, 0.55, 0.67, 0.87, and 1.02 microns. You will therefore need to output on your data on all of these. For some diagnostics, such as AOD, these are defined on a single level, but others, such as extinction, are a 3D field over the whole atmosphere. Therefore, to output the 3D aerosol extinction we need to make a new profile over all model (theta grid) levels, and all 6 pseudo levels.

We can now fill-in the values in the new domain profile accordingly, e.g.

- **dom_name:** e.g. D3DAR
- **iipl:** Variable derived on model theta levels (Charney-Phillips Grid) (2)
- **iilevs:** provide range [bottom and top level numbers]
- **iilev:** 1 (i.e. 20m level)
- **iilevt:** 85 (i.e. highest level)
- **plt:** Radiation bands for calculating aerosol optical depth (4)

This will then insert a new option below ($pslist$) to allow you to specify the bands you require.

- **pslist:** press the blue plus symbol (+) 5 times, and enter the numbers 1 to 6 in each of the boxes to output all AOT pseudo levels.
- **iopa:** Full model area (1)
- **imsk:** Land and sea points
- **im:** None (0)
- **iwt:** None
- **ts:** false

Usage profiles

Making new usage profiles is not as straightforward as for time and domain profiles, as these are used to output diagnostics to different output files (and possibly to climate mean files). To see what output streams are available, go to: $um \rightarrow namelist \rightarrow Model Input and Output \rightarrow Model Output Streams$. In the UKCA training suite, only $pp0$ is active.

To look at climate meaning, go to: $um \rightarrow namelist \rightarrow Model Input and Output \rightarrow Dumping and Meaning$. This is controlled by the logical $meaning.sequence$, which in the UKCA training suite is set to $false$ (i.e. no climate meaning). If this is set to true then various settings will need to be chosen, such as the reference time, the frequency of files, and which files to output.

To view the usage profiles, go to: $um \rightarrow namelist \rightarrow Model Input and Output \rightarrow STASH Requests and Profiles \rightarrow Usage Profiles$. This contains a list of existing domain profiles, with names like $upa_Elb3f000$ etc. If you look at the $upa$ namelist (by right-clicking on it and clicking view namelist) you can see that this corresponds with the $pp0$ output stream.

It is also possible to write diagnostics to a tag, which can then be read by other parts of the code. This is how UKCA couples to the UM, via tag=98. This can be seen in the $upuka$ usage profile.

When you take a copy of an existing configuration it is unlikely that you will need to make any changes to any (or make new) usage profiles. However, if you do want to make a new usage profile, you should:

1. Right-click anywhere on the list and click the blue plus symbol (+) named new section
   This will make a new blank line (usually labelled 1, the next 2 etc.) with a red X next to it.

2. Right-click this new line and click View namelist.

3. Fill-in the values as required for what you want to do.

4. When you have finished, you will need to run the When you have finished, you will need to run the $stashindices.TidyStashTransform$ macro by going to the Metadata $\rightarrow um$ drop-down menu. This will rename the new profile into the correct format.

Note that if you are making a new usage profile that is not associated with a tag, you may also need to make changes to the output streams. This could be quite involved.

Using your profiles

Once you have made your new profiles you should then use them in the usual way when outputting diagnostics, as covered in Tutorial 3.
When all your new STASH is set-up, you can use the Validator Macros to check if the names are correct, or if there are any unused STASH items etc. To run this, go to Metadata —> Check all Validator Macros from the drop-down menu. You can also check for fail-if or warn-if issues. However, these won’t warn of incorrect domain requests (i.e. requesting pressure-levels for diagnostics only defined on a single level), so some care should still be taken.

**Solution to Task 10.1**

You were given the task

- Output the following aerosol and radiation diagnostics to the UPA output stream. You should make a new time profile (called TRAD) to only output these on radiation timesteps. You will also need to make a new domain profile (called D3DAR) for s02i530 and s02i540, to output these on both model levels and pseudo levels.

<table>
<thead>
<tr>
<th>STASH Section</th>
<th>STASH Item</th>
<th>STASH Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>207</td>
<td>INCOMING SW RAD FLUX (TOA): ALL TSS</td>
</tr>
<tr>
<td>1</td>
<td>208</td>
<td>OUTGOING SW RAD FLUX (TOA)</td>
</tr>
<tr>
<td>2</td>
<td>205</td>
<td>OUTGOING LW RAD FLUX (TOA)</td>
</tr>
<tr>
<td>2</td>
<td>285</td>
<td>MINERAL DUST OPTICAL DEPTH IN RADN.</td>
</tr>
<tr>
<td>2</td>
<td>300</td>
<td>AITKEN MODE (SOLUBLE) OPTICAL DEPTH</td>
</tr>
<tr>
<td>2</td>
<td>301</td>
<td>ACCUM MODE (SOLUBLE) OPTICAL DEPTH</td>
</tr>
<tr>
<td>2</td>
<td>302</td>
<td>COARSE MODE (SOLUBLE) OPTICAL DEPTH</td>
</tr>
<tr>
<td>2</td>
<td>303</td>
<td>AITKEN MODE (INSOL) OPTICAL DEPTH</td>
</tr>
<tr>
<td>2</td>
<td>304</td>
<td>ACCUM MODE (INSOL) OPTICAL DEPTH</td>
</tr>
<tr>
<td>2</td>
<td>305</td>
<td>COARSE MODE (INSOL) OPTICAL DEPTH</td>
</tr>
<tr>
<td>2</td>
<td>585</td>
<td>MINERAL DUST ABS. OPTICAL DEPTH</td>
</tr>
<tr>
<td>2</td>
<td>240</td>
<td>AITKEN (SOLUBLE) ABS OPTICAL DEPTH</td>
</tr>
<tr>
<td>2</td>
<td>241</td>
<td>ACCUM (SOLUBLE) ABS OPTICAL DEPTH</td>
</tr>
<tr>
<td>2</td>
<td>242</td>
<td>COARSE (SOLUBLE) ABS OPTICAL DEPTH</td>
</tr>
<tr>
<td>2</td>
<td>243</td>
<td>AITKEN (INSOL) ABS OPTICAL DEPTH</td>
</tr>
<tr>
<td>2</td>
<td>244</td>
<td>ACCUM (INSOL) ABS OPTICAL DEPTH</td>
</tr>
<tr>
<td>2</td>
<td>245</td>
<td>COARSE (INSOL) ABS OPTICAL DEPTH</td>
</tr>
<tr>
<td>2</td>
<td>530</td>
<td>UKCA 3D AEROSOL EXTINCTION</td>
</tr>
<tr>
<td>2</td>
<td>540</td>
<td>CLASSIC 3D AEROSOL EXTINCTION</td>
</tr>
</tbody>
</table>

and were given the hint

- Remember to output your aerosol diagnostics on the AOT pseudo levels using the DIAGAOT profile.

For a working Rose suite that has completed this task, please see

- **ARCHER:** u-as292@62651
  - vm: u-as297@62631

The specific Rose changes made are:

- **ARCHER:** https://code.metoffice.gov.uk/trac/roses-u/changeset/62651/a/s/2/9/2/trunk
  - vm: https://code.metoffice.gov.uk/trac/roses-u/changeset/62631/a/s/2/9/7/trunk

The specific Rose changes made are:

**ARCHER:**

--- app/um/rose-app.conf

```
--- app/um/rose-app.conf       (revision 60289)
+++ app/um/rose-app.conf       (revision 62651)
@@ -2744,6 +2744,40 @
  precip_segment_size=32
  ukca_mode_seg_size=4
+
+  ![namelist:umstash_domain(d3dar_72578706)]
+  +dom_name='D3DAR'
+  !!iest=0
+  +ilevs=1
+  +imm=0
+  +imsk=1
+  +inth=0
+  +iopa=1
+  +iopl=2
+  +iisth=0
+  +iwt=0
+  +!!sml_ts=.false.
+  +levb=01
+  +!!levlst=0
+  +levt=85
+  +plt=4
+  +pslist=1,2,3,4,5,6
```

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_10
# UKCA Chemistry and Aerosol vn10.9 Tutorial 10 - UKCA

- **rlevlst=0**
- **spml_bot=0**
- **spml_ew=0**
- **spml_ns=0**
- **spml_top=0**
- **tlim=0**
- **telim=0**
- **nlim=0**
- **ts=.false.**
- **tslim=0**
- **tsnum=0**
- **ttlim=0**
- **ttlim=0**

```
[namelist:umstash_domain(dallrh_0496a967)]
  dom_name='DALLRH'
  iest=0
@@ -3025,6 +3059,22 @@
  tim_name='TALLTS'
  use_name='UPUKCA'
 +[namelist:umstash_streq(01207_ed72c304)]
+  dom_name='DIAG'
+  isec=1
+  item=207
+  package=
+  tim_name='TRAD'
+  use_name='UPA'
+  [namelist:umstash_streq(01208_83140cd8)]
+  dom_name='DIAG'
+  isec=1
+  item=208
+  package=
+  tim_name='TRAD'
+  use_name='UPA'
+[namelist:umstash_streq(01235_3511dd9f)]
  dom_name='DIAG'
  isec=1
@@ -3033,14 +3083,142 @@
  tim_name='TALLTS'
  use_name='UPUKCA'
- [namelist:umstash_streq(02301_0f7c5f4a)]
+ [namelist:umstash_streq(02205_357bf644)]
+  dom_name='DIAG'
+  isec=2
+  item=205
+  package=
+  tim_name='TRAD'
+  use_name='UPA'
+  [namelist:umstash_streq(02240_d97aaab7)]
+  dom_name='DIAGAOT'
+  isec=2
+  item=240
+  package=''
+  tim_name='TRAD'
+  use_name='UPA'
+  [namelist:umstash_streq(02241_8cda3169)]
+  dom_name='DIAGAOT'
+  isec=2
+  item=241
+  package=''
+  tim_name='TRAD'
+  use_name='UPA'
+  [namelist:umstash_streq(02242_91e371db)]
+  dom_name='DIAGAOT'
+  isec=2
+  item=242
+  package=''
+  tim_name='TRAD'
+  use_name='UPA'
+  [namelist:umstash_streq(02243_21bc5f11)]
```

---

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_10
+dom_name='DIAGAOT'
+isec=2
+item=243
+package=''
+tim_name='TRAD'
+use_name='UPA'
+
+[namelist:umstash_streq(02244_32fe0790)]
+dom_name='DIAGAOT'
+isec=2
+item=244
+package=''
+tim_name='TRAD'
+use_name='UPA'
+
+[namelist:umstash_streq(02245_da00b6ef)]
+dom_name='DIAGAOT'
+isec=2
+item=245
+package=''
+tim_name='TRAD'
+use_name='UPA'
+
+[namelist:umstash_streq(02285_1d9800f0)]
+dom_name='DIAGAOT'
+isec=2
+item=285
+package=''
+tim_name='TRAD'
+use_name='UPA'
+
+[namelist:umstash_streq(02300_8b9907b5)]
+dom_name='DIAGAOT'
+isec=2
+item=300
+package=''
+tim_name='TRAD'
+use_name='UPA'
+
+[namelist:umstash_streq(02301_b8711d23)]
+dom_name='DIAGAOT'
+isec=2
+item=301
+package='UKCA Testing'
+tim_name='T3HMN'
+package=''
+tim_name='TRAD'
+use_name='UPA'
+
+[namelist:umstash_streq(02302_420d0ec7)]
+dom_name='DIAGAOT'
+isec=2
+item=302
+package=''
+tim_name='TRAD'
+use_name='UPA'
+
+[namelist:umstash_streq(02303_ad5c3af4)]
+dom_name='DIAGAOT'
+isec=2
+item=303
+package=''
+tim_name='TRAD'
+use_name='UPA'
+
+[namelist:umstash_streq(02304_8c1869b6)]
+dom_name='DIAGAOT'
+isec=2
+item=304
+package=''
+tim_name='TRAD'
+use_name='UPA'
+
+[namelist:umstash_streq(02305_9ecd020a)]
+dom_name='DIAGAOT'
+isec=2
+item=305
+package=''
+tim_name='TRAD'
+use_name='UPA'

These differences can be found in the file /home/ukca/Tutorial/vn10.9/worked_solutions/Task10.1/Task10.1_rose.patch on PUMA.
+!!l_spml_ts=.false.
+!levb=01
+!!levlst=0
+!levt=85
+plt=4
+pslist=1,2,3,4,5,6
+!!levlst=0
+!!spml_bot=0
+!!spml_em=0
+!!spml_nw=0
+!!spml_top=0
+!!tblim=0
+!!tlimr=0
+!!tlim0=0
+!!ts=.false.
+!!tslim=0
+!!tsnum=0
+!!ttlim=0
+!!ttlimr=0
+!!twlim=0
+ [namelist:umstash_domain(dallrh_0496a967)]
  dom_name='DALLRH'
  !lest=0
@@ -3617,6 +3651,22 @@
  tim_name='TALLTS'
  use_name='UPUKCA'
  [namelist:umstash_streq(01207_ed72c304)]
+  +dom_name='DIAG'
+  +isec=1
+  +item=207
+  +package=
+  +tim_name='TRAD'
+  +use_name='UPA'
+  +
+  [namelist:umstash_streq(01208_83140cd8)]
+  +dom_name='DIAG'
+  +isec=1
+  +item=208
+  +package=
+  +tim_name='TRAD'
+  +use_name='UPA'
+  
+  [namelist:umstash_streq(02205_357bf644)]
+  +dom_name='DIAG'
+  +isec=1
+  +item=205
+  +package=
+  +tim_name='TRAD'
+  +use_name='UPA'
+  
+  [namelist:umstash_streq(02240_d97aaab7)]
+  +dom_name='DIAGAOT'
+  +isec=2
+  +item=240
+  +package=''
+  +tim_name='TRAD'
+  +use_name='UPA'
+  
+  [namelist:umstash_streq(02241_0cda3169)]
+  +dom_name='DIAGAOT'
+  +isec=2
+  +item=241
+  +package=''
+  +tim_name='TRAD'
+  +use_name='UPA'
+  
+  [namelist:umstash_streq(02242_91e371db)]
+  +dom_name='DIAGAOT'
+  +isec=2
http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_10

---

```plaintext
[namelist:umstash_streq(02243_21bc5f11)]
+dom_name='DIAGAOT'
+isec=2
+item=243
+package=''
+tim_name='TRAD'
+use_name='UPA'

[namelist:umstash_streq(02244_32fe0790)]
+dom_name='DIAGAOT'
+isec=2
+item=244
+package=''
+tim_name='TRAD'
+use_name='UPA'

[namelist:umstash_streq(02245_da00b6ef)]
+dom_name='DIAGAOT'
+isec=2
+item=245
+package=''
+tim_name='TRAD'
+use_name='UPA'

[namelist:umstash_streq(02285_1d9800f0)]
+dom_name='DIAGAOT'
+isec=2
+item=285
+package=''
+tim_name='TRAD'
+use_name='UPA'

[namelist:umstash_streq(02300_8b9907b5)]
+dom_name='DIAGAOT'
+isec=2
+item=300
+package=''
+tim_name='TRAD'
+use_name='UPA'

[namelist:umstash_streq(02301_b8711d23)]
+dom_name='DIAGAOT'
+isec=2
+item=301
+package='UKCA Testing'
+tim_name='T3HMN'
+package=''
+tim_name='TRAD'
+use_name='UPA'

[namelist:umstash_streq(02302_420d0ec7)]
+dom_name='DIAGAOT'
+isec=2
+item=302
+package=''
+tim_name='TRAD'
+use_name='UPA'

[namelist:umstash_streq(02303_ad5c3af4)]
+dom_name='DIAGAOT'
+isec=2
+item=303
+package=''
+tim_name='TRAD'
+use_name='UPA'

[namelist:umstash_streq(02304_8c1869b6)]
+dom_name='DIAGAOT'
+isec=2
+item=304
+package=''
+tim_name='TRAD'
+use_name='UPA'

[namelist:umstash_streq(02305_9ecd020a)]
```

---

9/21
If you open the .pa file in Xconv, you should see the following additional fields:

0    96  72  1   2  field200: INCOMING SW RAD FLUX (TOA): ALL TSS
1    96  72  1   2  field201: OUTGOING SW RAD FLUX (TOA)
2    96  72  1   2  olr: OUTGOING LW RAD FLUX (TOA)
3    96  72  6   2  unspecified: Stash code = 2240
4    96  72  6   2  unspecified: Stash code = 2241
5    96  72  6   2  unspecified: Stash code = 2242
6    96  72  6   2  unspecified: Stash code = 2243
7    96  72  6   2  unspecified: Stash code = 2244
8    96  72  6   2  unspecified: Stash code = 2245
9    96  72  6   2  unspecified: MINERAL DUST OPTICAL DEPTH IN RADN.
10   96  72  6   2  unspecified: AITKEN MODE (SOLUBLE) OPTICAL DEPTH
11   96  72  6   2  unspecified: ACCUM MODE (SOLUBLE) OPTICAL DEPTH
12   96  72  6   2  unspecified: COARSE MODE (SOLUBLE) OPTICAL DEPTH
13   96  72  6   2  unspecified: AITKEN MODE (INSOL) OPTICAL DEPTH
14   96  72  6   2  unspecified: ACCUM MODE (INSOL) OPTICAL DEPTH
15   96  72  6   2  unspecified: COARSE MODE (INSOL) OPTICAL DEPTH

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_10
Sample output from this task can be found at /work/n02/n02/ukca/Tutorial/vn10.9/sample_output/Task10.1/atmosa_pa19810901_00 on ARCHER.

**Task 10.2: Calculate aerosol optical depth**

**TASK 10.2:** Calculate the aerosol optical depth at 0.55 microns on the second radiation timestep.

<table>
<thead>
<tr>
<th>Hint</th>
<th>[hide]</th>
</tr>
</thead>
<tbody>
<tr>
<td>The UM calculate optical depth diagnostics on 6 pseudo levels corresponding to 0.38, 0.44, 0.55, 0.67, 0.87, and 1.02 microns, therefore 0.55 microns is pseudo level 3.</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** You will need to use the CLASSIC mineral dust optical depth diagnostic, as the configuration used in these tutorials does not use modal dust. Likewise, the insoluble accumulation and coarse mode diagnostics will be zero as these modes are not used in the configuration used here.

**Python script**

To calculate the total aerosol optical depth at 0.55 microns, you should sum up the contribution from the different aerosol components.

On ARCHER, an example python script to do this has been provided at /work/n02/n02/ukca/Tutorial/vn10.9/sample_output/Task10.2/write_AOD.py:

```python
#!/usr/bin/env python

# This file is part of the UKCA Tutorials:
# http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_Tutorials_at_vn10.9
# Copyright (C) 2017 University of Cambridge
# This is free software: you can redistribute it and/or modify it under the
terms of the GNU Lesser General Public License as published by the Free
Software Foundation, either version 3 of the License, or (at your option)
any later version.
# It is distributed in the hope that it will be useful, but WITHOUT ANY
# WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR
# A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more
details.
# Written by N. Luke Abraham 2017-12-11 <nla27@cam.ac.uk>
# preamble
import iris
import iris.time
fname='/work/n02/n02/ukca/Tutorial/vn10.9/sample_output/Task10.1/atmosa_pa19810901_00'
# constraint on time to get 2nd radiation timestep
tconstr=iris.Constraint(time=lambda cell: cell.point.hour == 2)
# load all AOD components at 0.55 micron
# must use this way of loading to account for constraint on time
with iris.FUTURE.context(cell_datetime_objects=True):
aod=iris.load(fname,[
    iris.Constraint(pseudo_level=3) & iris.AttributeConstraint(STASH='m01s02i285') & tconstr,
    iris.Constraint(pseudo_level=3) & iris.AttributeConstraint(STASH='m01s02i300') & tconstr,
    iris.Constraint(pseudo_level=3) & iris.AttributeConstraint(STASH='m01s02i301') & tconstr,
    iris.Constraint(pseudo_level=3) & iris.AttributeConstraint(STASH='m01s02i302') & tconstr,
    iris.Constraint(pseudo_level=3) & iris.AttributeConstraint(STASH='m01s02i303') & tconstr,
    iris.Constraint(pseudo_level=3) & iris.AttributeConstraint(STASH='m01s02i304') & tconstr,
    iris.Constraint(pseudo_level=3) & iris.AttributeConstraint(STASH='m01s02i305') & tconstr])
# make cube to store total AOD
aodsum=aod[0].copy()
```

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_10
Solution to Task 10.2

0.55 micron AOD calculated from the component aerosol AOD diagnostics.

You were asked to

- Calculate the aerosol optical depth at 0.55 microns on the second radiation timestep.

and were given the hint

- The UM calculate optical depth diagnostics on 6 pseudo levels corresponding to 0.38, 0.44, 0.55, 0.67, 0.87, and 1.02 microns, therefore 0.55 microns is pseudo level 3.

You should make use of the python script provided to do this.

Sample output from this task can be found in the `/work/n02/n02/ukca/Tutorial/vn10.9/sample_output/Task10.2/` directory on ARCHER, containing the following:

- Task102_AOD.nc
- write_AOD.py

Task 10.3: Calculate the single-scattering albedo

**TASK 10.3:** Calculate the single-scattering albedo at 0.55 microns on the second radiation timestep, defined as:

\[
1 - \left( \frac{\text{Absorption Aerosol Optical Depth}}{\text{Aerosol Optical Depth}} \right)
\]

**Python script**

To calculate the single-scattering albedo at 0.55 microns, you should sum up the contribution to AAOD and AOD from the different aerosol components.

On ARCHER, an example python script to do this has been provided at

/work/n02/n02/ukca/Tutorial/vn10.9/sample_output/Task10.3/write_SSA.py

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_10
#!/usr/bin/env python

# This file is part of the UKCA Tutorials:
# http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_Tutorials_at_vn10.9
# Copyright (C) 2017 University of Cambridge
# This is free software: you can redistribute it and/or modify it under the
# terms of the GNU Lesser General Public License as published by the Free
# Software Foundation, either version 3 of the License, or (at your option)
# any later version.
# It is distributed in the hope that it will be useful, but WITHOUT ANY
# WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR
# A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more
details.
# You find a copy of the GNU Lesser General Public License at <http://www.gnu.org/licenses/>.
# Written by N. Luke Abraham 2017-12-11 <nla27@cam.ac.uk>

# preamble
import iris
import iris.time

fname='/work/n02/n02/ukca/Tutorial/vn10.9/sample_output/Task10.1/atmosa.pa19810901_00'

tconstr=iris.Constraint(time=lambda cell: cell.point.hour == 2)

# load all AOD & AAOD components at 0.55 micron
# must use this way of loading to account for constraint on time
with iris.FUTURE.context(cell_datetime_objects=True):
aod=iris.load(fname, iris.Constraint(pseudo_level=3) & iris.AttributeConstraint(STASH='m01s02i285') & tconstr, iris.Constraint(pseudo_level=3) & iris.AttributeConstraint(STASH='m01s02i300') & tconstr, iris.Constraint(pseudo_level=3) & iris.AttributeConstraint(STASH='m01s02i301') & tconstr, iris.Constraint(pseudo_level=3) & iris.AttributeConstraint(STASH='m01s02i302') & tconstr, iris.Constraint(pseudo_level=3) & iris.AttributeConstraint(STASH='m01s02i303') & tconstr, iris.Constraint(pseudo_level=3) & iris.AttributeConstraint(STASH='m01s02i304') & tconstr, iris.Constraint(pseudo_level=3) & iris.AttributeConstraint(STASH='m01s02i305') & tconstr)
aaod=iris.load(fname, iris.Constraint(pseudo_level=3) & iris.AttributeConstraint(STASH='m01s02i585') & tconstr, iris.Constraint(pseudo_level=3) & iris.AttributeConstraint(STASH='m01s02i240') & tconstr, iris.Constraint(pseudo_level=3) & iris.AttributeConstraint(STASH='m01s02i241') & tconstr, iris.Constraint(pseudo_level=3) & iris.AttributeConstraint(STASH='m01s02i242') & tconstr, iris.Constraint(pseudo_level=3) & iris.AttributeConstraint(STASH='m01s02i243') & tconstr, iris.Constraint(pseudo_level=3) & iris.AttributeConstraint(STASH='m01s02i244') & tconstr, iris.Constraint(pseudo_level=3) & iris.AttributeConstraint(STASH='m01s02i245') & tconstr)

# make cube to store total AOD
aodsum=aod[0].copy()
# add-up components

# make cube to store total AAOD
aaodsum=aaod[0].copy()
# add-up components

# calculate single-scattering albedo
ssa=aodsum.copy()
ssa.data = 1.0 - (aaodsum.data/aodsum.data)
# rename
ssa.rename('single_scattering_albedo_in_air_due_to_ambient_aerosol_particles')
# remove unlimited dimension when writing to netCDF
iris.FUTURE.netcdf_no_unlimited=True
# output to netCDF
iris.save(ssa, 'Task103_SSA.nc', netcdf_format='NETCDF3_CLASSIC')

Solution to Task 10.3
0.55 micron single-scattering albedo.

You were asked to

- Calculate the single-scattering albedo at 0.55 microns on the second radiation timestep, defined as:

\[
1 - \left( \frac{\text{Absorption Aerosol Optical Depth}}{\text{Aerosol Optical Depth}} \right)
\]

You should use the python script provided to do this.

Sample output from this task can be found in the `/work/n02/n02/ukca/Tutorial/vn10.9/sample_output/Task10.3/` directory on ARCHER, containing the following:

- Task103_SSA.nc
- write_SSA.py

**Task 10.4: Calculate the top of the atmosphere net downward radiative flux**

**TASK 10.4:** Calculate the net downward top of the atmosphere radiative flux on the second radiation timestep.

**Python script**

To calculate the net TOA downward radiative flux, you should sum up the outgoing contributions from shortwave and longwave radiation, and take this away from the incoming shortwave radiative flux.

On ARCHER, an example python script to do this has been provided at

/work/n02/n02/ukca/Tutorial/vn10.9/sample_output/Task10.4/write_TOA.py:

```
#!/usr/bin/env python
#
# This file is part of the UKCA Tutorials:
# http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_Tutorials_at_vn10.9
#
# Copyright (C) 2017 University of Cambridge
#
# This is free software: you can redistribute it and/or modify it under the
# terms of the GNU Lesser General Public License as published by the Free
# Software Foundation, either version 3 of the License, or (at your option)
# any later version.
#
# It is distributed in the hope that it will be useful, but WITHOUT ANY
# WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR
# A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more
# details.
#
# You find a copy of the GNU Lesser General Public License at <http://www.gnu.org/licenses/>.
#
# Written by N. Luke Abraham 2017-12-11 <nla27@cam.ac.uk>
```

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_10
# preamble
import iris
import iris.time
fname='/work/n02/n02/ukca/Tutorial/vn10.9/sample_output/Task10.1/atmosa.pa19810901_00'
# constraint on time to get 2nd radiation timestep
tconstr=iris.Constraint(time=lambda cell: cell.point.hour == 2)
# load all TOA components at 0.55 micron
# must use this way of loading to account for constraint on time
with iris.FUTURE.context(cell_datetime_objects=True):
    isw=iris.load_cube(fname,[iris.AttributeConstraint(STASH='m01s01i207') & tconstr])
    osw=iris.load_cube(fname,[iris.AttributeConstraint(STASH='m01s01i208') & tconstr])
    olw=iris.load_cube(fname,[iris.AttributeConstraint(STASH='m01s02i205') & tconstr])
# make cube to store net downward TOA flux
toa=isw.copy()
# add-up components
toa.data=isw.data - (osw.data + olw.data)
toa.rename('toa_net_downward_radiative_flux')
# remove unlimited dimension when writing to netCDF
iris.FUTURE.netcdf_no_unlimited=True
# output to netCDF
iris.save(toa,'Task104_TOA.nc',netcdf_format='NETCDF3_CLASSIC')

Solution to Task 10.4

![Net downward TOA radiative flux](image)

You were asked to

- Calculate the net downward top of the atmosphere radiative flux on the second radiation timestep.

You should use the python script provided to do this.

Sample output from this task can be found in the `/work/n02/n02/ukca/Tutorial/vn10.9/sample_output/Task10.4/` directory on ARCHER, containing the following:

```
Task104_TOA.nc
write_TOA.py
```

**Task 10.5: Calculate aerosol optical depth from the 3D aerosol extinction**

**Task 10.5:** Using the 3D aerosol extinction, calculate the 0.55 micron aerosol optical depth on the second radiation timestep, and compare this to your AOD from Task 10.2.
Hint
You will need to include contributions from both UKCA and CLASSIC (due to the dust scheme).
Remember to correctly calculate the grid-cell heights.

Python script

To calculate the AOD from the aerosol extinction, you will need to integrate this in the column. To do this you should first multiply by the height of each grid-cell before summing-up.

On ARCHER, an example python script to do this has been provided at
/work/n02/n02/ukca/Tutorial/vn10.9/sample_output/Task10.5/calc_AOD.py:

```python
#!/usr/bin/env python
# This file is part of the UKCA Tutorials:
# http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_Tutorials_at_vn10.9
# Copyright (C) 2017 University of Cambridge
# This is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.
# It is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.
# You find a copy of the GNU Lesser General Public License at <http://www.gnu.org/licenses/>.
# Written by N. Luke Abraham 2017-12-11 <nla27@cam.ac.uk>

# preamble
import iris
import iris.time
import iris.analysis
import numpy as np

fname='/work/n02/n02/ukca/Tutorial/vn10.9/sample_output/Task10.1/atmosa.pa19810901_00'
# constraint on time to get 2nd radiation timestep
tconstr=iris.Constraint(time=lambda cell: cell.point.hour == 2)
# load orography to enable correct calculation of level heights
orog=iris.load_cube('/work/n02/n02/hum/ancil/atmos/n48e/orography/globe30/v1/qrparm.orog',iris.AttributeConstraint(STASH='m01s00i033'))
# load all extinction components at 0.55 micron
# must use this way of loading to account for constraint on time
with iris.FUTURE.context(cell_datetime_objects=True):
    ukca=iris.load_cube(fname,[
        iris.Constraint(pseudo_level=3) & iris.AttributeConstraint(STASH='m01s02i530') & tconstr])
    classic=iris.load_cube(fname,[
        iris.Constraint(pseudo_level=3) & iris.AttributeConstraint(STASH='m01s02i540') & tconstr])
# Calculate the correct height of each cell
# add the orography as an auxiliary coordinate
auxcoord=iris.coords.AuxCoord(orog.data,standard_name=str(orog.standard_name),long_name="orography",var_name="oro"
# added in to lat/lon (ht=0,lat=1,lon=2)
ukca.add_aux_coord(auxcoord,(1,2,))
# now calculate the correct altitude above sea-level
ukca.add_aux_factory(iris.aux_factory.HybridHeightFactory(delta=ukca.coord("level_height"),sigma=ukca.coord("sigma"),orography)
# now create the 'altitude' derived coordinate
ukca.add_aux_factory(ukca.coord("altitude"))
# now calculate the height from the bounds
bounds = ukca.coord('altitude').bounds[0] - ukca.coord('altitude').bounds[0]
# multiply by the height of each cell
ukca.data = ukca.data * bounds
classic.data = classic.data * bounds
# now sum up the column
ukca_int=ukca.collapsed('model_level_number',iris.analysis.SUM)
classic_int=classic.collapsed('model_level_number',iris.analysis.SUM)
# add together
aod=ukca_int.copy()```

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_10
On ARCHER, an example python script to calculate the difference from this AOD and the AOD calculated in Task10.2 is also provided at /work/n02/n02/ukca/Tutorial/vn10.9/sample_output/Task10.5/diff_AOD_methods.py:

```python
#!/usr/bin/env python
# This file is part of the UKCA Tutorials:
# http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_Tutorials_at_vn10.9
# Copyright (C) 2017  University of Cambridge
# This is free software: you can redistribute it and/or modify it under the
# terms of the GNU Lesser General Public License as published by the Free
# Software Foundation, either version 3 of the License, or (at your option)
# any later version.
# It is distributed in the hope that it will be useful, but WITHOUT ANY
# WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR
# A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more
# details.
# You find a copy of the GNU Lesser General Public License at <http://www.gnu.org/licenses/>.
# Written by N. Luke Abraham 2017-12-11 <nla27@cam.ac.uk>

# preamble
import iris

dname='./Task102_AOD.nc'
cname='./Task105_AOD.nc'
diag=iris.load_cube(dname)
calc=iris.load_cube(cname)

# difference the fields
calc.data=calc.data - diag.data

# remove unlimited dimension when writing to netCDF
iris.FUTURE.netcdf_no_unlimited=True

# output to netCDF
iris.save(calc,'Task105_AOD_diff.nc',netcdf_format='NETCDF3_CLASSIC')
```

Solution to Task 10.5

```python
aod.data = ukca_int.data + classic_int.data
# rename
aod.rename('atmosphere_optical_thickness_due_to_aerosol')

# remove unlimited dimension when writing to netCDF
iris.FUTURE.netcdf_no_unlimited=True

# output to netCDF
iris.save(aod,'Task105_AOD.nc',netcdf_format='NETCDF3_CLASSIC')
```
You were asked to

- **Using the 3D aerosol extinction**, calculate the 0.55 micron aerosol optical depth on the second radiation timestep, and compare this to your AOD from Task 10.2.

and were given the hints

- You will need to include contributions from both UKCA and CLASSIC (due to the dust scheme).
- Remember to correctly calculate the grid-cell heights.

You should use the python script provided to do this.

**Note** that there will be some small differences between these two fields, especially around coastlines.

Sample output from this task can be found in the `/work/n02/n02/ukca/Tutorial/vn10.9/sample_output/Task10.5/` directory on ARCHER, containing the following:

- Task105_AOD.nc
- Task105_AOD_diff.nc
- calc_AOD.py
- diff_AOD_methods.py

**Task 10.6: Calculate the difference in aerosol impacts when sec_org is no longer formed from ALICE**

**TASK 10.6:** You should now remove the formation of **Sec_Org** from the chemical reaction added in Task 6.1, giving

$$\text{ALICE} + \text{OH} \rightarrow \text{BOB}$$

and assess the impact this has on the aerosol diagnostics calculated in the previous tasks, 10.2, 10.3, and 10.4.

**Hint**

Remember to remove Sec_Org from the diagnostics as well as the reaction.

**Note:** as this a very short run (that is still spinning-up) and you will only be considering a single timestep, the results will be incredibly noisy.

**Code changes required to modify a reaction**

Modifying a reaction is very similar to adding a new one, although simpler as you are only changing what is already there. The exact routines that will need to be altered will depend on the reaction, but likely include:

- **ukca_chem_master.F90**

If you added any special code for the reaction(s) to **asad_bimol.F90**, **asad_trimol.F90**, **ukca_hetero_mod.F90**, **asad_hetero.F90**, **ukca_phot2d.F90**, or **ukca_strat_update.F90** it will need to be altered accordingly. The location of the reaction in the complete list is usually found by searching on the reactants and products, and if these change then the lookup will fail unless it is altered here too.
If there are any diagnostics to output the reactions you are modifying, you may need to alter this specification as well if you have changed the reactants and/or products.

Solution to Task 10.6

You were asked to

- You should now remove the formation of \texttt{Sec\_Org} from the chemical reaction added in Task 6.1, giving
  \[
  \text{ALICE + OH} \rightarrow \text{BOB}
  \]

and assess the impact this has on the aerosol diagnostics calculated in the previous tasks, 10.2, 10.3, and 10.4.

and were given the hint

- Remember to remove \texttt{Sec\_Org} from the diagnostics as well as the reaction.

You should use the above python scripts provided to do this.

For a working Rose suite that has completed this task, please see

- ARCHER: u-as292@62669
The specific Rose changes made are:

**ARCHER:**

```
vm: https://code.metoffice.gov.uk/trac/roses-u/changeset/62632/a/s/2/9/7/trunk
```

The specific UM changes made are:

```
vm: https://code.metoffice.gov.uk/trac/roses-u/changeset/62669/a/s/2/9/2/trunk
```

These differences can be found in the file `/home/ukca/Tutorial/vn10.9/worked_solutions/Task10.6/Task10.6_rose.patch` on PUMA.

Sample output from this task can be found in the `/work/n02/n02/ukca/Tutorial/vn10.9/sample_output/Task10.6/` directory on ARCHER, containing the following:

- Task106_AOD.nc
- Task106_AOD_diff.nc
- Task106_SSA.nc

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_10
Checklist

☐ Make new time and domain profiles as needed.
☐ Run the TidyStashTransform transform macro.
☐ Output your diagnostics in STASH at: um → namelist → Model Input and Output → STASH Requests and Profiles → STASH Requests.
☐ When adding new aerosol diagnostics, remember to output on pseudo levels if appropriate.
☐ Run the TidyStashTransform transform macro again for the diagnostic requests.
☐ Save your suite.
☐ In the roses/[SUITE-ID] directory, run fcm commit to commit your changes to the repository.
☐ When modifying chemical reactions, remember to make changes for diagnostics and special cases, as well as altering ukca_chem_master.F90.
☐ The Iris python library can be used to process and visualise Unified Model output.

Tutorial 11


- This page was last modified on 3 January 2018, at 11:59.
UKCA Chemistry and Aerosol vn10.9 Tutorial 11

From UKCA

UKCA Chemistry and Aerosol Tutorials at vn10.9

Contents

1. What you will learn in this Tutorial
2. Before you start this Tutorial
3. Create a ticket
4. Make a branch
5. Merge in Initial Changes
6. Run rose-stem
   - 6.1 Monsoon2
   - 6.2 umdp3_check
   - 6.3 fcm_make_vm.gnu.um_safe_noomp
   - 6.4 rose_analyze_vm.n48.ukca_eg_noomp.atmos_kgo
7. Adding new namelist input
   - 7.1 ukca_option_mod.F90
   - 7.2 ukca_light.F90
   - 7.3 rose-meta/um-atmos/HEAD/rose-meta.conf
   - 7.4 rose-meta/um-atmos/version109_110.py
   - 7.5 All changes
8. Make a test branch and test it with rose-stem
9. Documentation
   - 9.1 Ticket
   - 9.2 UMDP
      - 9.2.1 Virtual Machine Head of Trunk branch
      - 9.2.2 umdp84_app_control_vars.tex
      - 9.2.3 umdp84_emissions.tex
      - 9.2.4 UM version
10. Next Steps
    - 10.1 Ticket Summary
    - 10.2 SciTech Review
    - 10.3 Code Review
11. Checklist

What you will learn in this Tutorial

In this tutorial you will go through a worked example showing you how to commit code to the UM trunk.

You should also open the working practices for the UM (https://code.metoffice.gov.uk/trac/um/wiki/working_practices) page on MOSRS and read through that as well.

This can only be completed on the Virtual Machine, as rose-stem does not currently work on ARCHER.

Before you start this Tutorial

Before you start this tutorial you will need to have a working Virtual Machine where known good output (or KGO) has been installed for the vs_n48.ukca_eg_noomp. You can do this by running the command

```
rose stem --group=vm_n48.ukca_eg_noomp -S GENERATE_KGO=true
```

within a vanilla copy of the vn10.9 trunk. When this runs for the first time the two KGO tasks will fail, but when this happens the KGO is installed. You just need to re-trigger these two tasks to run again, and they will succeed and the suite will stop.


Create a ticket

You will need to fill-in several headings. These can all be changed later if needed.

- **Summary**: Give a short description of what the change is for.
- **Description**: Give a longer and more detailed description.
- **Type**: Putting enhancement is usually fine here.
- **Milestone**: This should be the UM (or Mule) version that you are targeting. If you don't know a particular version leave this as **Hopefully**, and if you aren't targeting a version then you can put this as **Not for Builds**. In this example please use **Not for Builds**.
- **Severity**: Usually changes will be minor or significant, depending on if results are changed. Usually wholesale and trivial aren't used very often. This doesn't matter too much, as the Code Reviewer will often change this during their review.
- **Keywords**: When working on UKCA, please include UKCA and SC0138 (this is to make it easy to search for UKCA tickets)

Once you have created it, you should **Modify** it and **start work** and then **click submit**. This will change the ownership to you and change the ticket statute to in_progress. You will now be able to find your ticket on your view tickets (https://code.metoffice.gov.uk/trac/um/query) page.
Tickets are important as they track the change through the MOSRS system. While initially "owned" by you, when you submit your code for review it will be passed to someone else. They will then mark on the ticket when it passes (or fails) their review and pass it on to the next stage prior to commit (or back to you to fix/clarify things). When the code is eventually committed it will be classed as "closed" and passed back to you.

It is best to keep the front page of the ticket relatively clear, as it will be used to pass information between yourself and the reviewers as the ticket progresses. You can always make sub-pages linked from the ticket to hold more information if needed.

Make a branch

Please make a branch as covered in Tutorial 4, e.g.

```
fcm branch-create --type dev -k ticket_number your_branch_name fcm:um.x_tr@vn10.9
```

and then check-out your branch by

```
fcm checkout fcm:um.x_br/dev/userid/vn10.9 your_branch_name
```

Merge in Initial Changes

You should merge-in branch dev/lukeabraham/vn10.9_UKCA_Worked_Example by changing into the top-level directory of your branch and doing

```
fcm merge fcm:um.x_br/dev/lukeabraham/vn10.9_UKCA_Worked_Example@47427
```

(Note: the revision number 47427 is important here) and then `fcm commit` your branch.

The code that you are checking in can be found here: https://code.metoffice.gov.uk/trac/um/changeset/47427

```
Index: src/atmosphere/UKCA/ukca_light.F90
===================================================================
--- src/atmosphere/UKCA/ukca_light.F90 (revision 47420)
+++ src/atmosphere/UKCA/ukca_light.F90 (revision 47427)
@@ -241,27 +241,60 @@
!         3 from cloud base to 2 above top
!         KLT is the level above cloud top
-dpcg = ppress(1) - ppress(jniv)
dpcc = ppress(jniv) - ppress(klt)
+IF (l_ukca_linox_logp) THEN
+   ! DO EVERYTHING LINEARLY IN LOG(PRESSURE)
+   ! sanity check to prevent dpcg==0
+   IF (jniv <= 1) jniv=2
+   dpcg = LOG(ppress(1)) - LOG(ppress(jniv))
dpcc = LOG(ppress(jniv)) - LOG(ppress(klt))
! ...construct L-NOx profile in kg(N)/gridcell/s
! ...first cloud-to-ground L-NOx profiles (kg(N)/gridcell/s)
! IF ((jniv-1) == 1) THEN
!   anox(1) = acgnox
! ELSE
!   DO k = 1,jniv-1
!      anox(k) = acgnox * ((LOG(ppress(k))=LOG(ppress(k+1)))/dpcg)
!   END DO
! END IF
! ...
! ...then cloud-to-cloud L-NOx profiles (kg(N)/gridcell/s)
IF (LOG(ppress(jniv)) <= LOG(ppress(klt))) THEN
! jniv (level of the 500hPa level) is above the
! cloud-top-height. In this case, put all C2C N
! into the cloud-top level.
anox(klt-1) = anox(klt-1) + accnox
ELSE
! jniv is greater than the cloud-top-height
! Note: anox(k) is also on the RHS of this equation
DO k = jniv,klt-1
   anox(k) = anox(k) + accnox * ( (LOG(ppress(k))=LOG(ppress(k+1)))/dpcc )
END DO
END IF
ELSE
! .not. l_ukca_linox_logp
! DO EVERYTHING LINEARLY IN PRESSURE
-dpcg = ppress(1) - ppress(jniv)
dpcc = ppress(jniv) - ppress(klt)
! ...construct L-NOx profile in kg(N)/gridcell/s
! ...first cloud-to-ground L-NOx profiles (kg(N)/gridcell/s)
IF ((jniv-1) == 1) THEN
```

You can find an example of this here: um:#3639 (https://code.metoffice.gov.uk/trac/um/ticket/3639).
The aim of this change is to allow the code to either redistribute lightning NOx emissions vertically either linearly in pressure (the current default) or linearly in LOG(pressure) (the new change).

Take a look through this code. Can you spot any issues that you think it might have?

Run rose-stem

You should now run this through rose-stem by running the command

```
rose stem --group=vm_n48_ukca_eg_noomp,umdp3_check
```

The vm_n48_ukca_eg_noomp will run the code through one of the UKCA jobs, and the umdp3_check will test for coding standard compliance.

Rose-stem will fail. How does it fail? What are the error messages? What can you do to fix them?

Monsoon2

If you have access to Monsoon2, you should be able to run equivalent test to this by using the following rose-stem command

```
rose stem --group=meto_xc40_n48_ukca_eg_omp_noios_gnu,umdp3_check -S PROJECT="your-monsoon-project" -S HOST_XC
```

This will run the Met Office (meto) equivalent to the virtual machine rose-stem job. The xc40 indicates that it runs on the Cray XC40 architecture, rather than elsewhere (e.g. on a GNU/Linux system).

Note that as all the Met Office rose-stem tests are available, running --group=developer, --group=ukca, or any one of the larger groups will take a long time to complete due to the reduced size of the system. It is best to limit the number of tests to the minimum for this exercise.

umdp3_check

The error message (in the job.out file) is:

```
The following files have failed the UMDF3 compliance tests:
File src/atmosphere/UKCA/ukca_light.F90 :
Line longer than 80 characters: ' anox(k) = anox(k) + accnox * ( (LOG(press(k)) - LOG(press(k+1))) / dpcc)
```

The following code snippet is highlighted:

```fortran
+  anox(1) = acgnox
+  ELSE
+    DO k = 1,jniv-1
+      anox(k) = acgnox * ((ppress(k)-ppress(k+1))/dpcg)
+    END DO
+  END IF
+  ! ...then cloud-to-cloud L-NOx profiles (kg(N)/gridcell/s)
+  IF (ppress(jniv) <= ppress(klt)) THEN
+    anox(klt-1) = anox(klt-1) + accnox
+  ELSE
+    DO k = jniv,klt-1
+      anox(k) = accnox * ((ppress(k)-ppress(k+1))/dpcc)
+    END DO
+  END IF
+  END IF ! l_ukca_linox_logp
-  ! ...construct L-NOx profile in kg(N)/gridcell/s
-  ! first cloud-to-ground L-NOx profiles (kg(N)/gridcell/s)
-  IF (((jniv-1) == 1) THEN
-    anox(1) = acgnox
-  ELSE
-    DO k = 1,jniv-1
-      anox(k) = acgnox * ((ppress(k)-ppress(k+1))/dpcg)
-    END DO
-  END IF
-  ! ...then cloud-to-cloud L-NOx profiles (kg(N)/gridcell/s)
-  IF (ppress(jniv) <= ppress(klt)) THEN
-    anox(klt-1) = anox(klt-1) + accnox
-  ELSE
-    DO k = jniv,klt-1
-      anox(k) = accnox * ((ppress(k)-ppress(k+1))/dpcc)
-    END DO
-  END IF
-  END IF
```

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_11
This is relatively easy to solve by adding a continuation line.

Index: src/atmosphere/UKCA/ukca_light.F90
--- src/atmosphere/UKCA/ukca_light.F90 (revision 47427)
+++ src/atmosphere/UKCA/ukca_light.F90 (revision 47436)
@@ -268,7 +268,8 @@
 ! jniv is greater than the cloud-top-height
 ! Note: anox(k) is also on the RHS of this equation
 DO k = jniv,klt-1
-         anox(k) = anox(k) + accnox * ( (LOG(ppress(k)) - LOG(ppress(k+1))) / dpcc )
+         anox(k) = anox(k) + accnox * 
+              ( (LOG(ppress(k)) - LOG(ppress(k+1))) / dpcc )
 END DO
 END IF
 ELSE ! .not. l_ukca_linox_logp

See https://code.metoffice.gov.uk/trac/um/changeset/47436

If you make this change and then fcm commit you can then re-run the umdp3_check task again (after stopping any still running suites).

rose stem --group=umdp3_check

You should find that it passes successfully.

fcm_make_vm_gnu_um_safe_noomp

The error is (in the job.err fail) is:

This can be easily fixed by adding the following at the top of the ukca_light.F90 routine:

Index: src/atmosphere/UKCA/ukca_light.F90
--- src/atmosphere/UKCA/ukca_light.F90 (revision 47436)
+++ src/atmosphere/UKCA/ukca_light.F90 (revision 47439)
@@ -138,6 +138,9 @@
 REAL :: fr_calib_fac ! model resolution calibration factor

 The logical is set to .false. above, meaning that the code will continue to use the old method.

You can now run rose-stem again:

rose stem --group=vm_n48_ukca_eg_noomp

Rose-stem will now pass all tests and complete successfully.

What happens if you set l_ukca_linox_logp = .true.?

Index: src/atmosphere/UKCA/ukca_light.F90
--- src/atmosphere/UKCA/ukca_light.F90 (revision 47439)
+++ src/atmosphere/UKCA/ukca_light.F90 (revision 47446)
@@ -139,7 +139,7 @@
 REAL :: fr_calib_fac ! model resolution calibration factor

 See https://code.metoffice.gov.uk/trac/um/changeset/47439/

 Note that the logical is set to .false. above, meaning that the code will continue to use the old method.

You can now run rose-stem again:

rose stem --group=vm_n48_ukca_eg_noomp

Rose-stem will now pass all tests and complete successfully.
! logical to select redistribution in pressure (F) or LOG(pressure) (T)
-LOGICAL, PARAMETER :: l_ukca_linox_logp = .false.
+LOGICAL, PARAMETER :: l_ukca_linox_logp = .true.

INTEGER(KIND=jpim), PARAMETER :: zhook_in = 0
INTEGER(KIND=jpim), PARAMETER :: zhook_out = 1

See https://code.metoffice.gov.uk/trac/um/changeset/47446

If you now make this change and re-run rose stem

rose stem --group=vm_n48_ukca_eg_noomp

what happens?

rose_ana_vm_n48_ukca_eg_noomp_ atmos_kgo

Now, the rose-ana task fails with the following error (in the job.out file):

```plaintext
[FAIL] File 1: /home/vagrant/umdir/standard_jobs/kgo/vm_n48_ukca_eg_1x2/vn10.9/atmosa.pa19810901_00
[FAIL] File 2: /home/vagrant/cylc-run/vn10.9_UKCA_Worked_Example/work/1/atmos_vm_n48_ukca_eg_noomp_1x2/atmosa.pa
[FAIL] Files DO NOT compare
[FAIL] * 0 differences in fixed_length_header (with 7 ignored indices)
[FAIL] * 387 field differences, of which 387 are in data
[FAIL] Compared 583/583 fields, with 196 matches
[FAIL] Maximum RMS diff as % of data in file 1: 129.036987416 (field 496)
[FAIL] Maximum RMS diff as % of data in file 2: 56.3431083041 (field 496)
```

i.e. the results have changed. When finished, rose-stem will generate a file, called trac.log, that can be found in the cylc-run directory of your branch. For the test above, this will show that it has failed one test, but passed the others.

You can see an example of this output here: https://code.metoffice.gov.uk/trac/um/wiki/ticket/3639/stemfail

The trac.log file is formatted so that when it is pasted onto the MOSRS Twiki system it will appear as a formatted table. This is because rose-stem output (in the form of the trac.log file) is required to complete a ticket for submission to the trunk. More on this will be discussed later.

It is important not to paste the contents of the trac.log file to the front page of the ticket, as the formatting makes this hard to read. If you want to keep a record of multiple rose-stem tests, you should make a new wiki page linked from the ticket and copy the information there.

Hard-coding a logical is not a good idea in this case, as it can only be changed by re-compiling the code. It is easy to forget to do this, and also many UM runs use standard binaries.

Adding new namelist input

A better solution is to introduce the logical l_ukca_linox_logp into the input namelist and allow it to be set by a user in Rose. To do this we must make several code changes.

**ukca_option_mod.F90**

All variables within the run_ukca namelist, that holds all the UKCA inputs, are declared in `ukca_option_mod.F90`. This module also contains the namelist itself, as well as some umPrint statements and handling for parallel calls. To add a new namelist variable you need to:

1. Add a declaration for it, e.g.
   ```f90
   LOGICAL :: l_ukca_linox_logp = .FALSE.
   ```
   All logicals must be set to .FALSE., character variables must be set to `VARIABLE NAME is unset` and integers and reals must be set to missing data values (defined as imd and rm in respectively).

2. Add the variable to the run_ukca namelist.

3. Print the value, e.g.
   ```f90
   WRITE(lineBuffer,'(A33,L1)') ' l_ukca_linox_logp = ', l_ukca_linox_logp
   CALL umPrint(lineBuffer,src='ukca_option_mod')
   ```

4. Add the variable to the my_namelist derived type.
5. On processor 0, save the value of the new variable within the my_nml variable (which is of type my_namelist).
6. After my_nml has been broadcast, on all other processors copy the value of the variable outside of the my_nml variable and into the variable itself. This then means that it can be used as expected in all routines.
7. If required, you may also need to add some checks on the variable to the check_run_ukca routine within this module, to ensure that the value (or value range) is correct.

**ukca_light.F90**

In `ukca_light.F90` we now need to stop declaring the variable at the top of the routine, and instead USE it from `ukca_option_mod.F90`, e.g.
USE ukca_option_mod, ONLY: l_ukca_linox_logp

rose-meta/um-atmos/HEAD/rose-meta.conf

We need to add a section here on the new variable to tell Rose how to treat it, e.g.

[namelist:run_ukca=l_ukca_linox_logp]
compulsory=true
description=When T, Lightning NOx emissions are distributed
          =vertically using LOG(pressure)
          =redistribute the Lightning NOx emissions in the vertical
          =linearly using LOG(pressure)
sort-key=b16

There needs to be help text (which will be shown by Rose when selecting help when clicking on the cog associated with the variable, as well as short description that is shown under the variable name to tell the user what it's for. The sort-key is used to order the variables in the Rose panel. For integer or real variables, it's possible to give an allowed range of values, or use radio buttons etc.

rose-meta/um-atmos/version109_110.py

As a new namelist input has been added (and similarly if one is removed or changed in some way), then an upgrade macro will need to be written to all the new variable to be automatically be inserted into Rose. When the UM goes from one version to another these macros are run sequentially to upgrade Rose to be able to use the new UM version, containing all the new (or changed) namelist items.

This macro takes the form of a short python function which inserts default values (formatted as strings), and looks like this:

class vn109_t3639(rose.upgrade.MacroUpgrade):
    ""
    Upgrade macro for ticket #3639 by Luke Abraham."
    BEFORE_TAG = "vn10.9"
    AFTER_TAG = "vn10.9_t3639"

    def upgrade(self, config, meta_config=None):
        ""
        Introduce logical to interpolate linearly in LOG(p)
        for the vertical redistribution of Lightning NOx.
        ""
        self.add_setting(config,"namelist:run_ukca",
                         "l_ukca_linox_logp",".false.")

        return config, self.reports

The name of the function is of the form vnXY_ticket_number.

All changes

Index: src/atmosphere/UKCA/ukca_light.F90
===================================================================
--- src/atmosphere/UKCA/ukca_light.F90 (revision 47446)
+++ src/atmosphere/UKCA/ukca_light.F90 (revision 47454)
@@ -57,6 +57,7 @@
 USE parkind1,        ONLY: jprb, jpim
 USE parcons_mod,     ONLY: rad, deg
 USE ukca_constants,  ONLY: avc
+USE ukca_option_mod, ONLY: l_ukca_linox_logp

IMPLICIT NONE
@@ -138,9 +139,6 @@
 REAL :: ns_res_deg ! gridbox height in degrees
 REAL :: fr_calib_fac ! model resolution calibration factor

!-- logical to select redistribution in pressure (F) or LOG(pressure) (T)
-LOGICAL, PARAMETER :: l_ukca_linox_logp = .true.
 INTEGER(KIND=jpim), PARAMETER :: zhock_in = 0
 INTEGER(KIND=jpim), PARAMETER :: zhock_out = 1
 REAL(KIND=jprb) :: zhock_handle
Index: src/atmosphere/UKCA/ukca_option_mod.F90
===================================================================
--- src/atmosphere/UKCA/ukca_option_mod.F90 (revision 47446)
+++ src/atmosphere/UKCA/ukca_option_mod.F90 (revision 47454)

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_11

6/11
! T to pass columns to ASAD rather than theta_field
LOGICAL :: l_ukca_asad_columns = .FALSE.

!* T to use LOG(p) to distribute lightning NOx in the vertical
+LOGICAL :: l_ukca_linox_logp = .FALSE.

INTEGER :: chem_timestep = imdi         ! Chemical timestep in seconds for N-R
! and Offline oxidant schemes
INTEGER :: dts0 = 300                   ! Default Backward Euler timestep

! These are set in ukca_setup_chem_mod after the namelist is read
!! 1 ukca_use_background_aerosol
WRITE(lineBuffer,'(A33,L1)')' l_ukca_asad_columns = ', &
 l_ukca_asad_columns
+WRITE(lineBuffer,'(A33,L1)')' l_ukca_linox_logp = ', &
 +      l_ukca_linox_logp
CALL umPrint(lineBuffer,src='ukca_option_mod')

WRITE(lineBuffer,'(A33,L1)')' l_ukca_primsu = ',l_ukca_primsu
CALL umPrint(lineBuffer,src='ukca_option_mod')

! These are set in ukca_setup_chem_mod after the namelist is read
!! 1 ukca_use_background_aerosol
WRITE(lineBuffer,'(A33,L1)')' l_ukca_asad_columns = ', &
 l_ukca_asad_columns
+WRITE(lineBuffer,'(A33,L1)')' l_ukca_linox_logp = ', &
 +      l_ukca_linox_logp
CALL umPrint(lineBuffer,src='ukca_option_mod')

WRITE(lineBuffer,'(A33,L1)')' l_ukca_primsu = ',l_ukca_primsu
CALL umPrint(lineBuffer,src='ukca_option_mod')

my_nml % l_ukca_so2ems_expvolc = l_ukca_so2ems_expvolc
my_nml % l_ukca_quasinewton = l_ukca_quasinewton
my_nml % l_ukca_limit_nat = l_ukca_limit_nat
my_nml % l_ukca_linox_logp = l_ukca_linox_logp

! end of logicals
my_nml % jvspec_dir = jvspec_dir
my_nml % jvspec_file = jvspec_file

! these are set in ukca_setup_chem_mod after the namelist is read
my_nml % l_ukca_so2ems_expvolc = l_ukca_so2ems_expvolc
my_nml % l_ukca_quasinewton = l_ukca_quasinewton
my_nml % l_ukca_limit_nat = l_ukca_limit_nat
my_nml % l_ukca_linox_logp = l_ukca_linox_logp

! end of logicals
jvspec_dir = my_nml % jvspec_dir
jvspec_file = my_nml % jvspec_file

--- rose-meta/um-atmos/version109_110.py (revision 47446)
+++ rose-meta/um-atmos/version109_110.py (revision 47454)
@@ -17,15 +17,23 @@
+
--- rose-meta/um-atmos/version109_110.py (revision 47446)
+++ rose-meta/um-atmos/version109_110.py (revision 47454)
@@ -17,13 +17,21 @@

Index: rose-meta/um-atmos/version109_110.py
===================================================================
--- rose-meta/um-atmos/version109_110.py (revision 47446)
+++ rose-meta/um-atmos/version109_110.py (revision 47454)
@@ -17,13 +17,21 @@

Index: rose-meta/um-atmos/version109_110.py
===================================================================
--- rose-meta/um-atmos/version109_110.py (revision 47446)
+++ rose-meta/um-atmos/version109_110.py (revision 47454)
@@ -17,13 +17,21 @@

Index: rose-meta/um-atmos/version109_110.py
===================================================================
Index: rose-meta/um-atmos/HEAD/rose-meta.conf
===================================================================
--- rose-meta/um-atmos/HEAD/rose-meta.conf (revision 47446)
+++ rose-meta/um-atmos/HEAD/rose-meta.conf (revision 47454)
@@ -23534,6 +23534,16 @@
      sort-key=b15
      type=real
+([namelist:run_ukca=l_ukca_linox_logp]
 + compulsory=true
 + description=When T, Lightning NOx emissions are distributed
      =vertically using LOG(pressure)
 + help=When T, this logical makes the UKCA Lightning NOx routine
      =redistribute the Lightning NOx emissions in the vertical
 + sort-key=b16
 + type=logical
+[
+namelist:run_ukca=max_ageair_reset_height]
 compulsory=true
 description=Maximum height for resetting Age-of-air tracer values

See https://code.metoffice.gov.uk/trac/um/changeset/47454/

Make a test branch and test it with rose-stem

    fcm branch-create -k ticket_number --branch-of-branch -t test your_branch_name fcm:um.x_br/dev/your_MOSRS_username

You should then checkout this branch, cd into its top-level directory and run the command

    ./admin/rose-stem/update_all.py /home/vagrant/path/to/zXXXXX_your_branch_name --um=vn10.9_ticket_number

where XXXX is the last revision number of vn10.9_your_branch_name prior to you creating the test branch.

This python script will then go through and upgrade all the apps, which are different configurations of the UM used for testing, and insert the new namelist input and check the metadata.

A full list of the changes made in this case can be found here: https://code.metoffice.gov.uk/trac/um/changeset/47481

You should fcm commit your changes, and then run rose-stem again

    rose stem --group=vm_n48_ukca_eg_noomp,scripts

where the scripts group runs the umdp3_check as well as some other checks. This will now pass, and you can see in the trac.log (https://code.metoffice.gov.uk/trac/um/wiki/ticket/3639/stempass) file that everything has completed successfully.

The Virtual Machine set of standard tests can be found here: https://code.metoffice.gov.uk/trac/um/wiki/VirtualMachine/StandardJobs10.9

The Met Office set of standard tests can be found here: https://code.metoffice.gov.uk/trac/um/wiki/StandardJobs10.9

The minimum group that is needed to be checked is developer group at the Met Office. If you do not have access to Monsoon2 you will need to ask someone who does to run these tests, or ask someone with access to the internal Met Office network.

Documentation

There are two levels of documentation on the ticket and in the relevant documentation paper (UMDP). All changes must have a ticket description, and some may also require updates to one or model Unified Model documentation papers (UMDPs).

Ticket

For small changes, all that will be required is some text in the ticket Description, but for more involved changes a ticket details page will need to be added. To do this, modify the ticket to add the following at the end of the description:

    Upgrade a UM runtime app configuration.

    # Input your macro commands here

    # Introduce logical to interpolate linearly in LOG(p) for the vertical redistribution of Lightning NOx.

    self.add_setting(config,["namelist:run_ukca",
                           "l_ukca_linox_logp"],".false.")

    return config, self.reports
Save, and then click on the gray link. You will then be asked to create the page, and do so using the TicketDetails template from the drop-down menu. This will make a blank page containing the following:

```
= Ticket Details <#Ticket Nr> =
== Author: <name> ==
-----
== Branch ==
[log:main/branches/]
-----
== Documentation and Testing ==
Please add details here:
-----
```

You should add-in the information on the branch and the ticket etc., and then add details under the Documentation and Testing section. This should be as detailed as possible to describe what the ticket does. A blank example for this ticket/branch can be found here: https://code.metoffice.gov.uk/um/work/wiki/ticket/3639/TicketDetails

**UMDP**

Please also see https://code.metoffice.gov.uk/um/work/wiki/WorkingPractices/Documentation/UpdatingUMDPs

**Note:** Until UM vn11.0 it is not possible to generate UMDPs on the VM - this requires the change in #3612 (https://code.metoffice.gov.uk/um/work/ticket/3612), which has been committed at r47913 (https://code.metoffice.gov.uk/um/work/changeset/47913). This means it is possible to make a documentation branch from the head of the trunk (HoT) instead.

Small changes often do not require detailed documentation to be added - the ticket itself is usually sufficient. However, for changes to namelist variables (such as the change added here) the documentation must be updated. This does not need to be a big change, but the changes to the namelist need to be kept up-to-date. The documentation paper is a *living document* and so needs to reflect the state of the code that is being committed to the trunk.

A documentation branch is required for this, e.g.

```
fcm branch-create -k ticket_number your_documentation_branch_name fcm:um_doc.x_br@vn10.9
```

where `your_documentation_branch_name` will need to be different from `your_branch_name` to avoid confusion.

**Virtual Machine Head of Trunk branch**

To make a HoT branch on the VM the process is similar, e.g.

```
fcm branch-create -k ticket_number your_documentation_branch_name fcm:um_doc.x_br@HEAD
```

The name the branch created will now be something like rXXXXX_your_documentation_branch_name rather than vn10.9_your_documentation_branch_name. This is the only appreciable difference. If you wanted instead of HEAD you could also specify r47913 instead.

For the VM to be able to build UMDPs correctly you need to have created a VM after 6th December 2017 that includes pull request #12 (https://github.com/metomi/metomi-vms/pull/12) as this adds support for LaTeX. You may also need to run `apt-get` to install a PDF viewer such as evince or xpdf.

You should then checkout your documentation branch and make the changes required. For this change they are:

```
Index: doc/branches/dev/lukeabraham/vn10.9_UKCA_Worked_Example_Doc/source/084/umdp84_app_control_vars.tex
===================================================================
--- a/doc/branches/dev/lukeabraham/vn10.9_UKCA_Worked_Example_Doc/source/084/umdp84_app_control_vars.tex
+++ b/doc/branches/dev/lukeabraham/vn10.9_UKCA_Worked_Example_Doc/source/084/umdp84_app_control_vars.tex
@@ -93,4 +93,6 @@
 L\_UKCA\_ASAD\_COLUMNS & True to pass columns to ASAD Newton-Raphson solver \ 
 & rather than latitude-longitude slices. \\ 
+ L\_UKCA\_LINOX\_LOGP  & True to redistribute Lightning NOx \ 
+ & emissions vertically using log(pressure) \ 
\hline
\multicolumn{2}{|l|}{MODE Options} \\
```
A technical or scientific description of the change will also be required. Here this should be made to the source/084/umdp84_emissions.tex file, e.g.

Index: doc/branches/dev/lukeabraham/vn10.9_UKCA_Worked_Example_Doc/source/084/umdp84_emissions.tex

--- a/doc/branches/dev/lukeabraham/vn10.9_UKCA_Worked_Example_Doc/source/084/umdp84_emissions.tex
+++ b/doc/branches/dev/lukeabraham/vn10.9_UKCA_Worked_Example_Doc/source/084/umdp84_emissions.tex
@@ -265,4 +265,5 @@
 users to scale the derived Lightning NO$_x$ emissions by any factor between 0.0 and 5.0.
 *If the control logical L\_UKCA\_LINOX\_LOGP is true, lightning NO$_x$ emissions will be vertically redistribute linearly in pressure.

\subsection{Sea salt primary emissions}

**UM version**

Unless you are working on a head of trunk branch, you will also need to update the UM version number in the UMDP to show that the change is in line with the next version. This is done in the source/084/umdp84.tex file:

This must be done for all changes to UMDPs

The documentation branch containing these changes can be seen here: https://code.metoffice.gov.uk/trac/um/log/doc/branches/dev/lukeabraham/vn10.9_UKCA_Worked_Example_Doc

The UMDP containing all these changes can be found here: https://code.metoffice.gov.uk/trac/um/attachment/ticket/3639/Worked_Example-umdp_084.pdf

**Next Steps**

**Ticket Summary**

Once you've documented your ticket and made your UMDP, you will need to make a ticket summary prior to submitting the change to the trunk. A blank example can be found here: https://code.metoffice.gov.uk/trac/um/wiki/ticket/3639/TicketSummary

You will need to answer all the questions and append your final rose-stem output.

**SciTech Review**

When you have completed everything you will pass the ticket to someone else for Sci/Tech Review. This person may be the code owner of the (largest) section you are altering, or it may be someone else suitable who you have previously approached. They will then complete the Sci/Tech Review template and will either approve the change, or ask for further changes or clarification. If the latter, they will pass the ticket back to you, and when you're happy that you've answered all questions you should hand the ticket back to Sci/Tech Reviewer, and document the ticket, ticket details, or ticket summary with more information as appropriate.

When you pass the ticket to the Sci/Tech reviewer it is also a good idea to email the UM Systems Team at this point and ask for the name of the Code Reviewer, as the Sci/Tech Reviewer can pass it directly to them when the change is approved.

A blank Sci/Tech Review template can be found here: https://code.metoffice.gov.uk/trac/um/wiki/ticket/3639/SciTechReview

**Code Review**

Anyone with an SRS account can be a Sci/Tech Reviewer, but (generally) only members of the UM Systems Team can be Code Reviewers, as the Code Reviewer is the person who will commit the change to the trunk.

Once your change has passed Sci/Tech Review it will then be checked by the Code Reviewer for coding standards and consistency, whether the code change is linked to changes in other models, such as JULES, whether documentation has been made etc. A blank Code/System Review template can be seen here: https://code.metoffice.gov.uk/trac/um/wiki/ticket/3639/CodeSystemReview

The Code Reviewer may have further changes or questions which will also need to be answered prior to the code being accepted for the UM trunk. They will pass the ticket back to you, and you will need to make changes and then pass the ticket back to them before they will respond. It is important that the ownership of the ticket is changed to the correct person at each step.

Once the code change has been approved the status of the ticket will be changed to approved, and when the code has been committed it will be committed. Then the code change will be automatically be checked overnight in the standard testing suites, and if there are no problems the ticket will be set to closed a few days later. At that point the Code Reviewer will change the ownership of the ticket to you again.

Congratulations! You have successfully had a change committed to the trunk!

**Checklist**

http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_vn10.9_Tutorial_11
☐ Use rose-stem to test and develop your change.
☐ Remember to check for UMDBP3 code compliance.
☐ Document your ticket sufficiently.
☐ You may need to make a test branch.
☐ You may need to update UMDBPs. Remember to update the UM version number if you do.
☐ Remember to respond to all questions/corrections from your SciTech and Code reviewers.
☐ Remember to pass ownership of the ticket back to the appropriate reviewer when required.

UKCA Chemistry and Aerosol Tutorials at vn10.9

Written by Luke Abraham 2017


- This page was last modified on 26 January 2018, at 11:05.