NEAMS Workbench PyARC Application Local and Remote Setup

Robert A. Lefebvre, lefebvrera@ornl.gov

Argonne National Laboratory

November 21, 2019

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

Assumptions

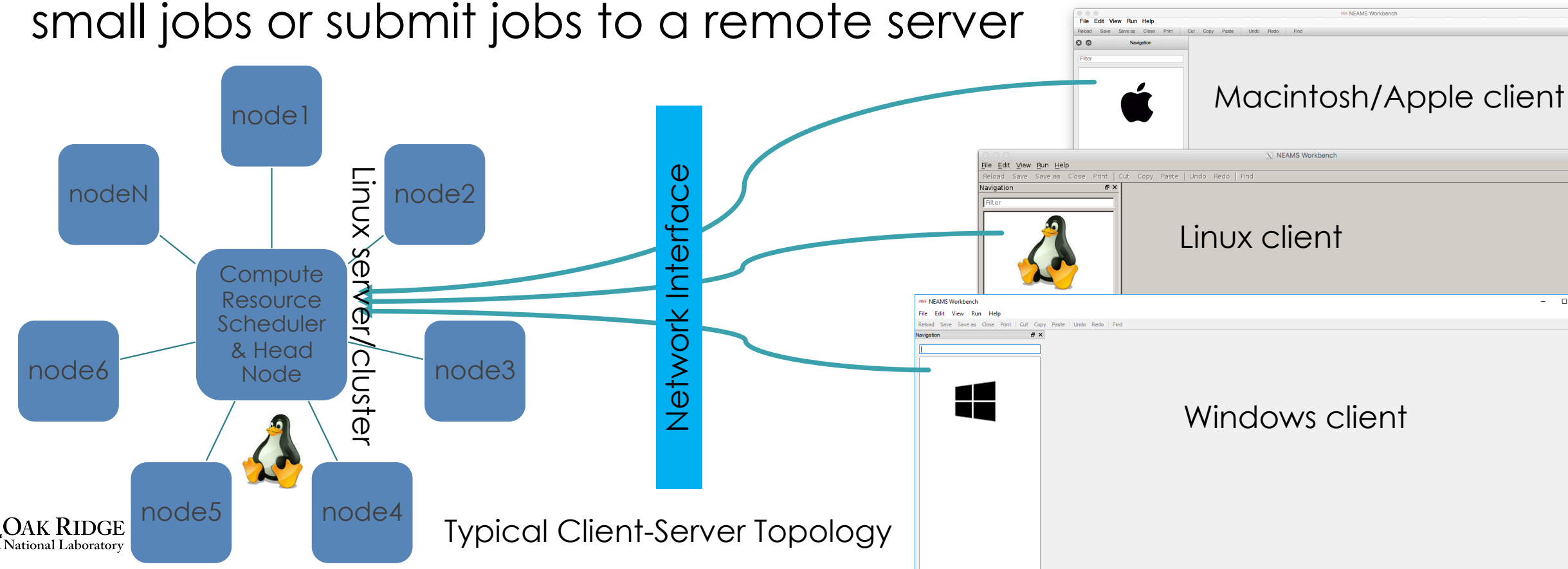
- The following slides assume
  - the NEAMS Workbench has already been installed
    - NEAMS Workbench is freely available from https://code.ornl.gov/neams-workbench/downloads
    - Installation notes are at https://code.ornl.gov/neams-workbench/downloads#install-neams-workbench-beta
  - the ARC code suite has been installed
  - PyARC has been configured to use the ARC code suite

Need assistance? Email nwb-help@ornl.gov

OAK RIDGE National Laboratory

Client vs. Server definition

- ‘Server’, a.k.a. a ‘cluster’, is where compute resources are available and jobs can be scheduled
- ‘Client’, a.k.a. a ‘local’ machine, is typically a laptop or desktop with limited resource that can be used to locally run small jobs or submit jobs to a remote server

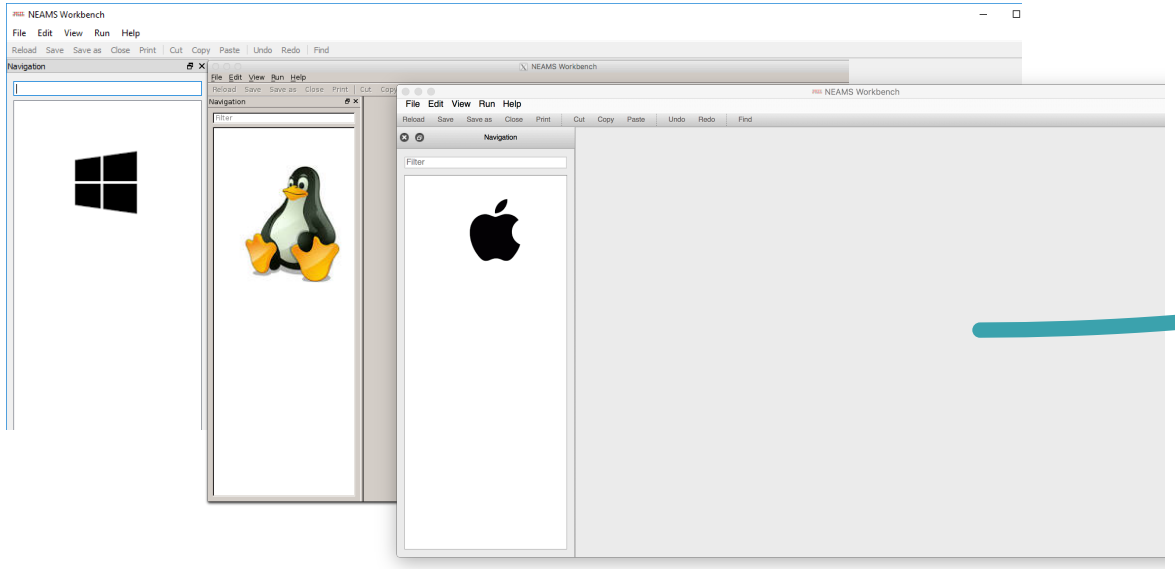


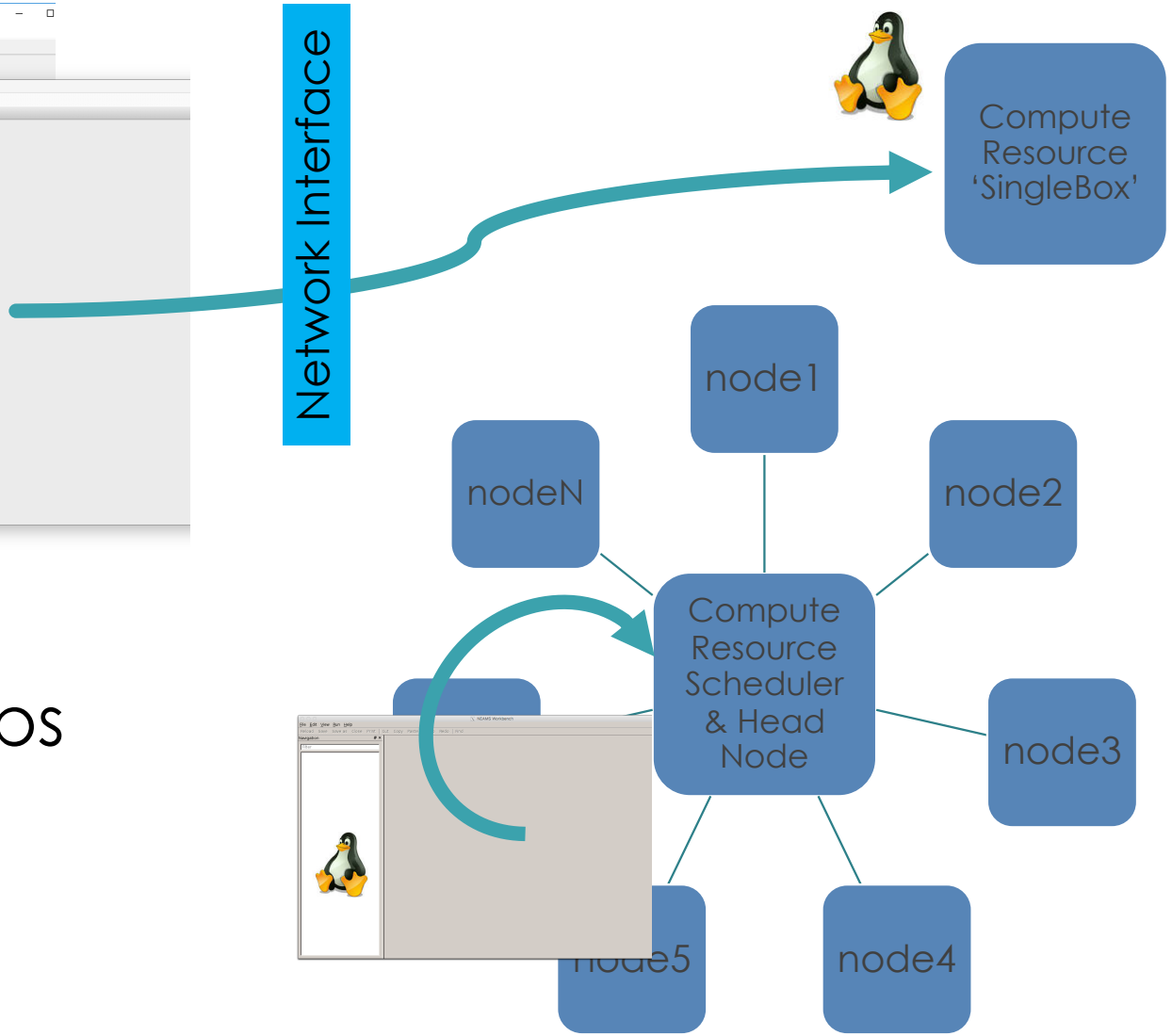
Typical Client-Server Topology

OAK RIDGE National Laboratory

Additional configurations

- Single-box – compute resource without a scheduler where jobs are submitted across the network





- Localhost – client is the cluster head node with scheduler from which jobs are submitted locally to the scheduler

Oak Ridge National Laboratory

NEAMS Workbench Linux Server Installation

1. Download/copy the Workbench-Linux.zip to the server
   - Secure copy from client to server – `scp Workbench-Linux.zip server:`
2. Unzip the deployment – `unzip Workbench-Linux.zip`
3. Untar the archive – `tar -xf Workbench-Linux.tar.gz`
4. Verify installation via the following
   1. Execute **`Workbench-Linux/rte/entry.sh`** which should result in Python terminal with no import errors
   2. Execute **`Workbench-Linux/bin/sonvalidjson`** which should result in a usage message (no library load errors)

Latest Linux distribution is available from https://code.ornl.gov/neams-workbench/downloads

NOTE: be sure to use the same client and server versions. Server installations can be shared between users as long as the administrating user executes the `Workbench-Linux/rte/entry.sh` to configure the necessary Python environment

OAK RIDGE National Laboratory

ARC Linux Server Installation

- Obtain the program executables and libraries from RSICC and Argonne National Laboratory
- Configure installation into the following standard

arc/bin
arc/bin/lib

Where arc/bin contains the program executables

```
dif3d_gamsor.x
dif3d_to_vtk.x
dif3d.x
ISOTXS_merge.x
mcc3.x
partisn.x
persent.x
rebus.x
twodant.x
```

If the executables are not already in this configuration one can use soft-links to properly configure. E.g., ln -s /path/to/dif3d dif3d.x

OAK RIDGE National Laboratory

Configure PyARC Linux Server Installation

- Create the NEAMS*-Linux/PyARC/**.pyarc.rc** file with the following content

```
[exec]
path = /projects/arc/bin
```

- Ensure the **path** property lists the location previously configured during the ARC Linux Server Installation step

OAK RIDGE National Laboratory

Local and Remote Application Setup Overview

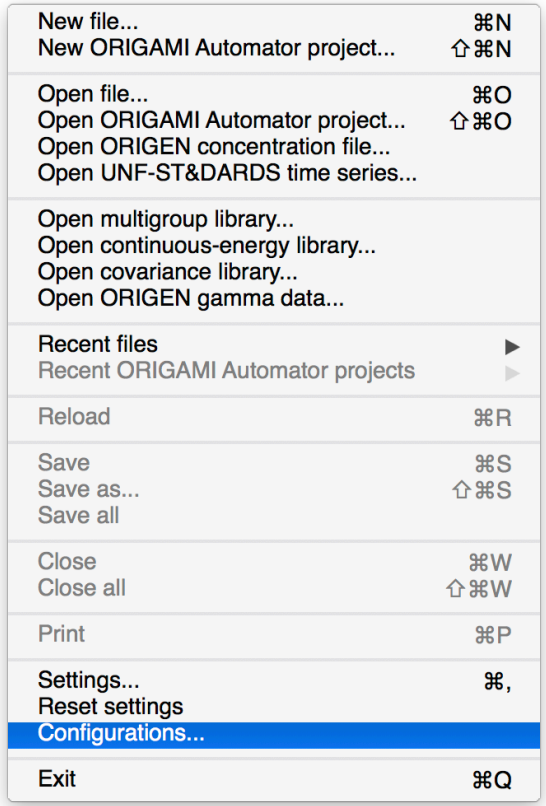
- Local application setup requires installation of the ARC code suite on the same machine as is the NEAMS Workbench
- Remote application setup requirements
  - Passwordless SSH* to remote **Linux** machine/cluster
  - PBS-based cluster scheduler
    - Remote application does support scheduler-less single-box
    - Tested with Torque/MAUI, IBM LSF, SLURM
  - Initial setup step using the NEAMS Workbench SetupRemote application*
  - Remote Linux machine installation of the application and the NEAMS Workbench

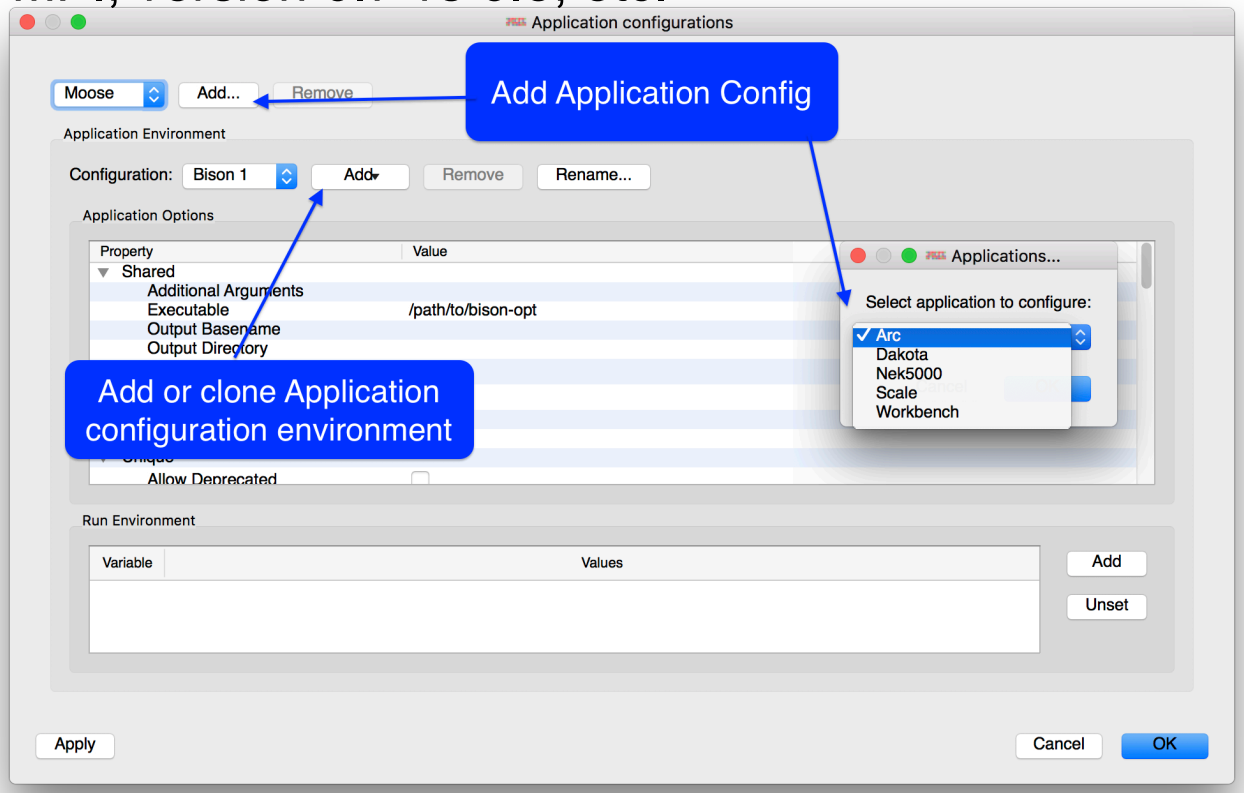
* See later slide instructions

OAK RIDGE National Laboratory

Run Configurations - Overview

- Application configurations are available under the **File>Configurations…**
- Create application run configuration
  - Connects MOOSE-BISON, Dakota, ARC, etc. to the NEAMS Workbench
- Allows creation of application variants/environments
  - Serial vs MPI, version 6.7 vs 6.8, etc.



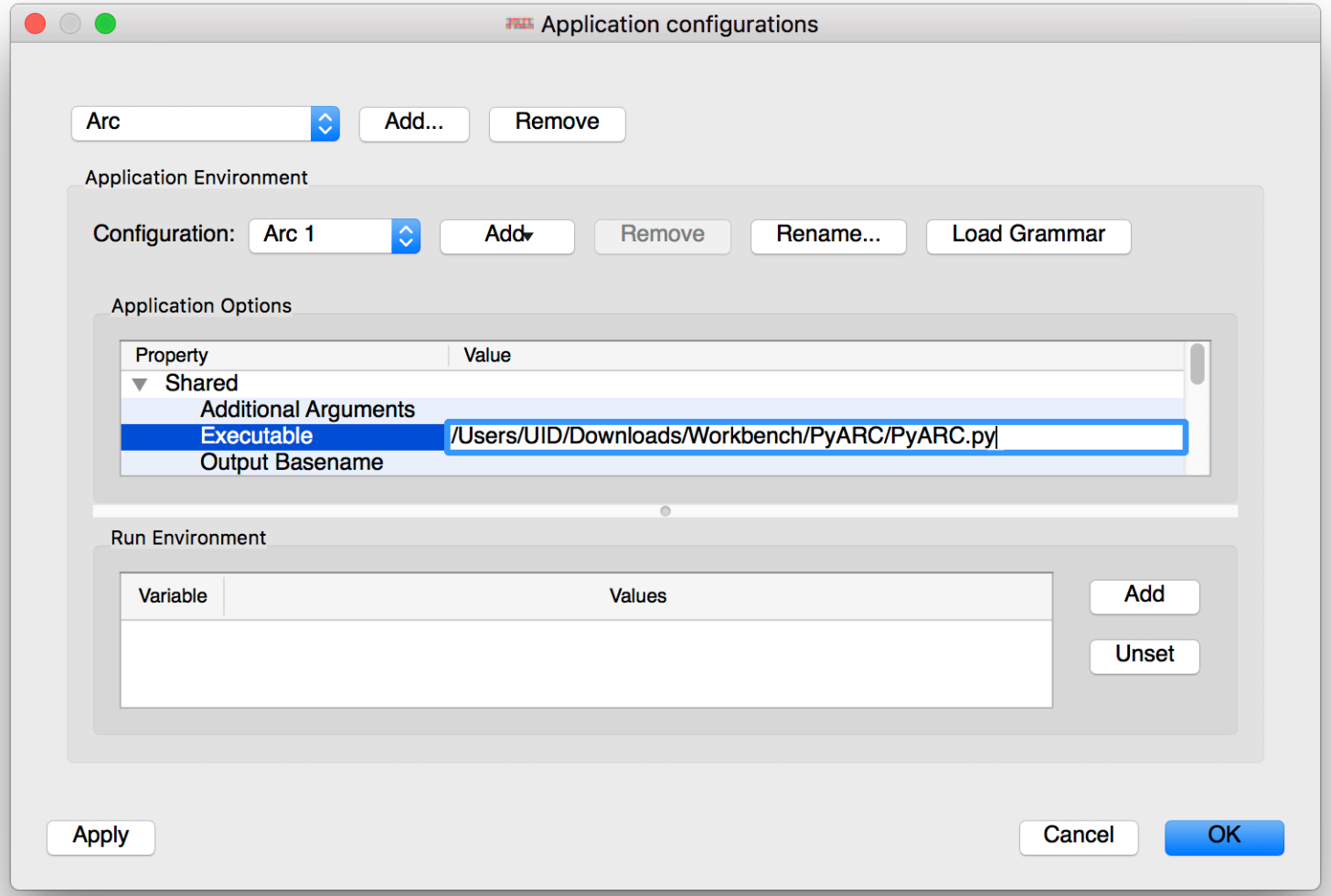


Oak Ridge National Laboratory

Local Configuration of ARC suite via PyARC

1. Open the Application Configuration dialog via **File>Configurations**
2. Select **Add… Arc**
3. Update the **Executable** property's value to be the absolute path to PyARC application script

Note: PyARC is deployed with the Workbench at Workbench/PyARC



Oak Ridge National Laboratory

Remote Run Configuration of PyARC Overview

1. Configure passwordless SSH between the client and the server
2. Using the client install of the NEAMS Workbench, do the following:
   a. Enable the NEAMS Workbench's Setup_Remote application
   b. Create the application-server Setup_Remote input
   c. Run application-server setup remote input to generate the NEAMS Workbench PyARC remote run configuration.
   d. Configure the PyARC_remote
   e. Open any PyARC input and select the PyARC remote run configuration to activate input validation, autocompletion, syntax highlighting and job launch capabilities
   f. Edit input as needed
   g. Use the PyARC remote run configuration to launch the PyARC input from your client to the server
   h. Wait for job to finish
   i. Open files associated with the input file (*.out, summary, *.vtk, etc.)

OAK RIDGE National Laboratory

Passwordless SSH Setup

- Required for remote run configurations
- Appends client machine's authorized public key to the user's list of authorized keys residing on the server. The following statement is sufficient to do so on **Linux** or **Mac** clients:

```
ssh server 'cat >> .ssh/authorized_keys && chmod 644 .ssh/authorized_keys' < ~/.ssh/id_rsa.pub
```

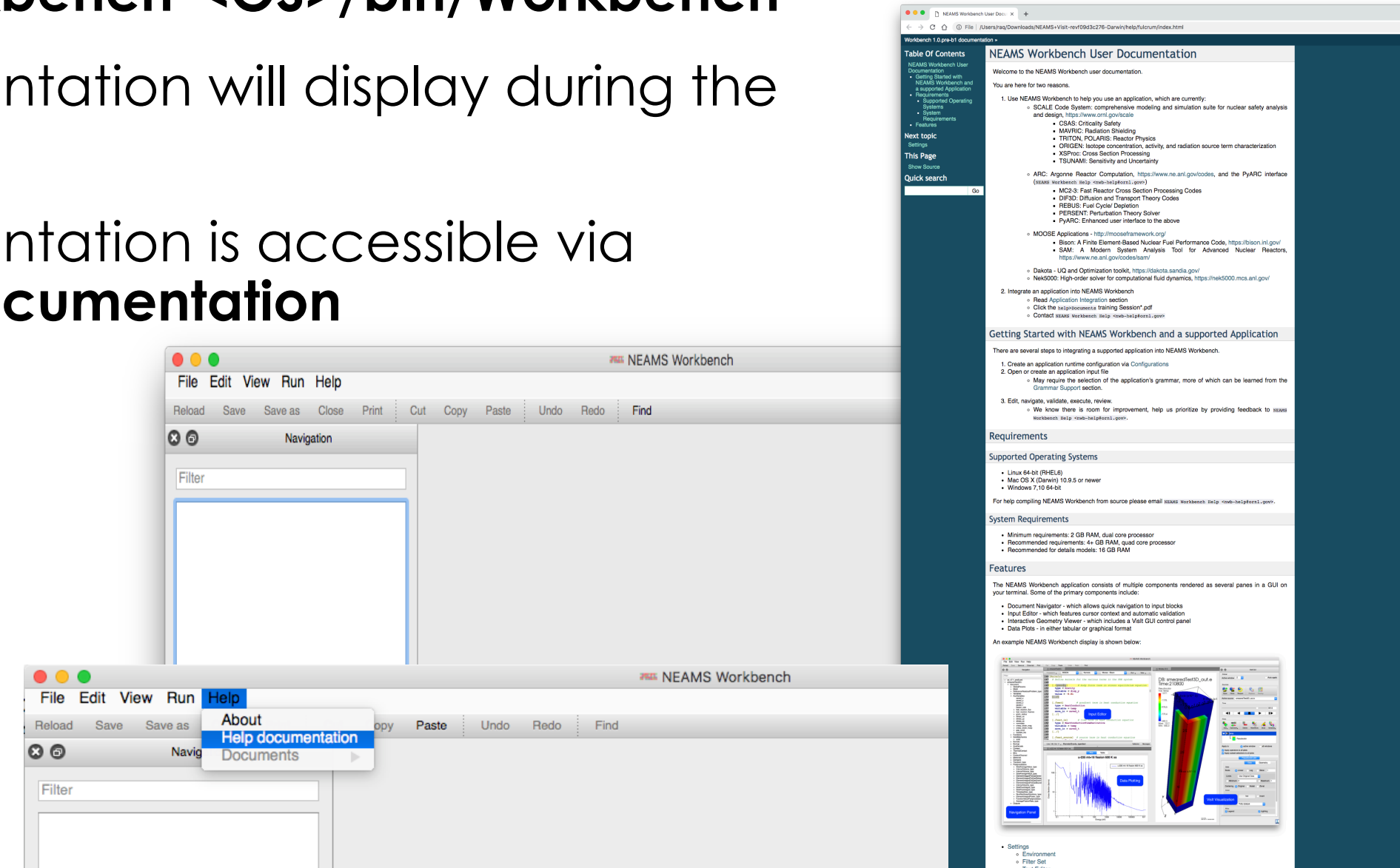
- Replace 'server' with the name of target server. Test the passwordless ssh by attempting to ssh to the server. You should no longer observe the need to specify a password.
- Assumes server:**.ssh** directory exists. If this does not exist, run **ssh-keygen** on the server to generate the SSH directory.
- On **Windows** clients use the **CMD** console program to execute

```
type .ssh\id_rsa.pub | ssh server "cat >> .ssh/authorized_keys && chmod 644 .ssh/authorized_keys"
```


OAK RIDGE National Laboratory

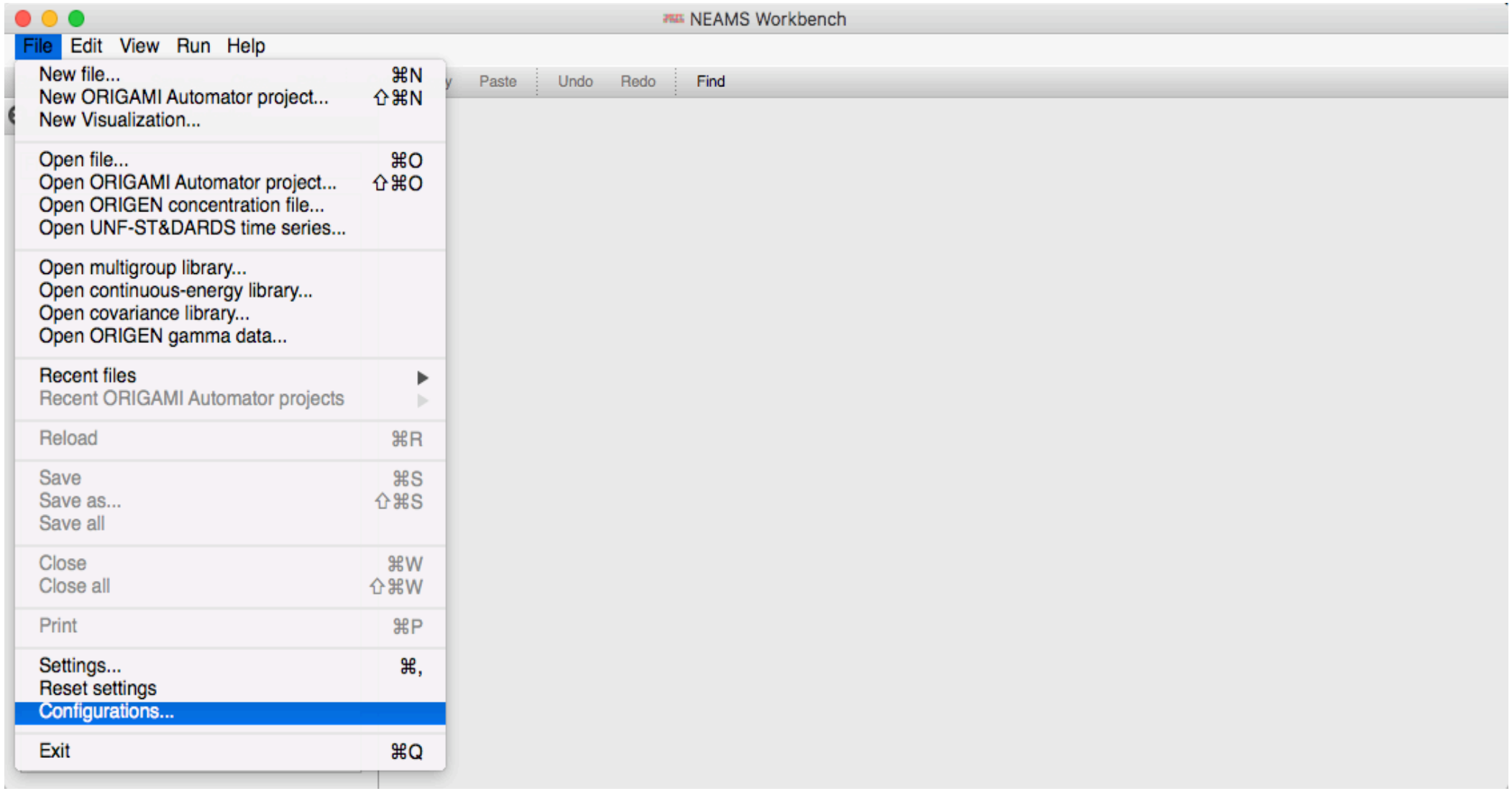
Start the client install of the NEAMS Workbench

- Execute **Workbench-<OS>/bin/Workbench**
- Help documentation will display during the first startup
- Help documentation is accessible via **Help>Help Documentation**



Configure the Setup Remote Application

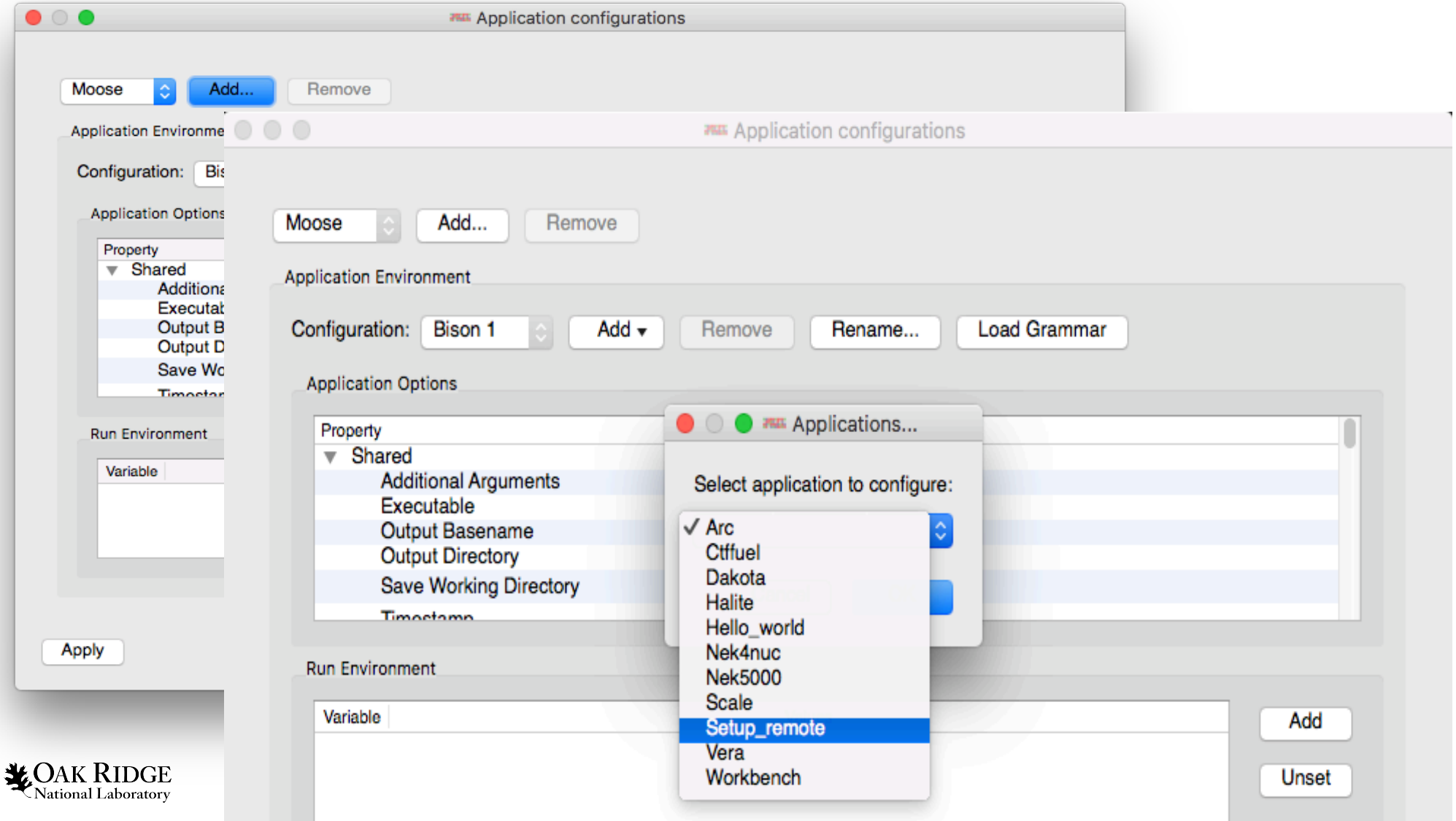
- Click **File>Configurations…**



OAK RIDGE National Laboratory

Configure the Setup Remote Application

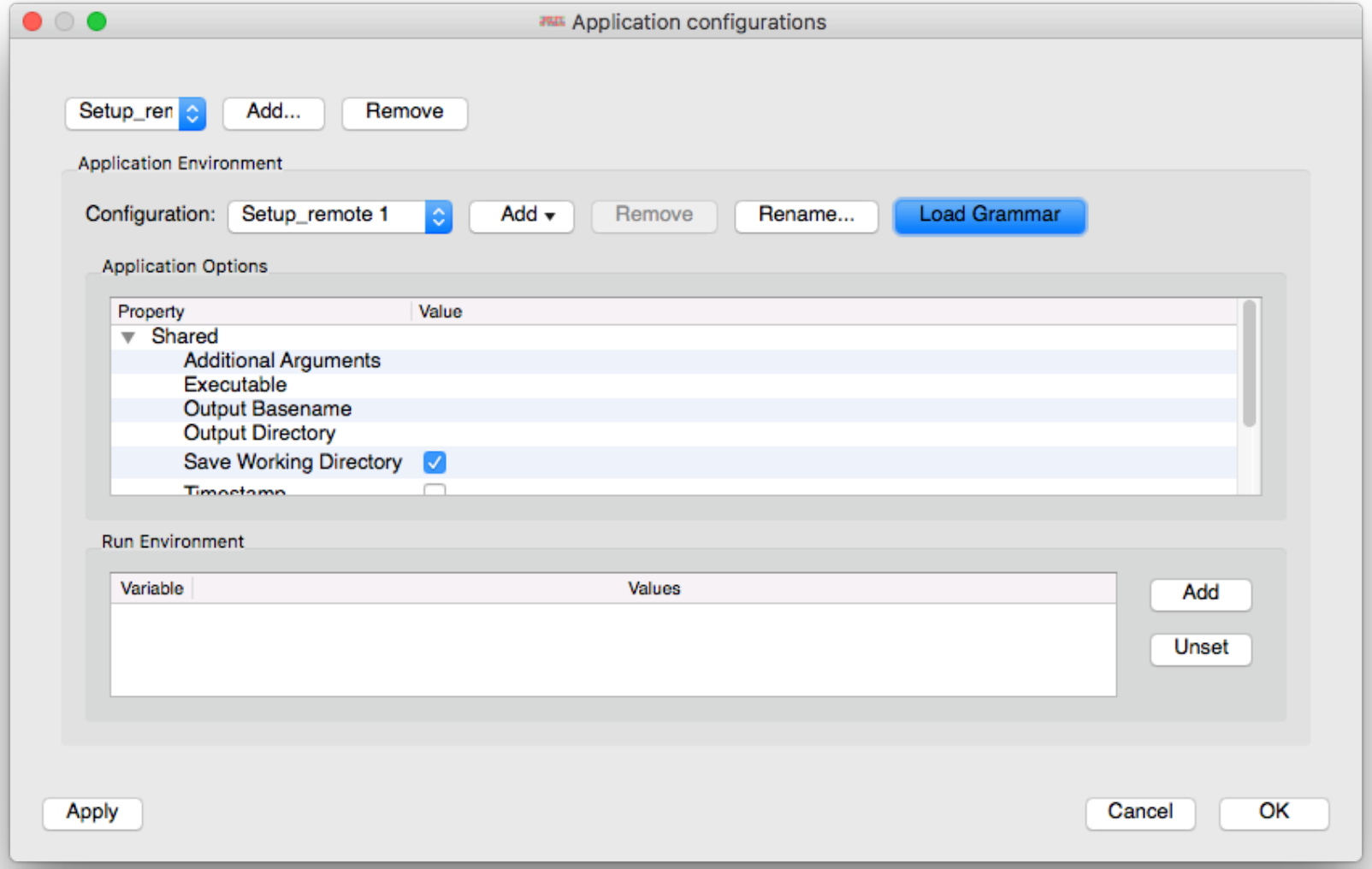
- Click **Add…** and select **Setup_remote** and click **OK**



Oak Ridge National Laboratory

Configure the Setup Remote Application

- Click **Load Grammar**, **Apply**, and **OK.** This enables SetupRemote input creation.

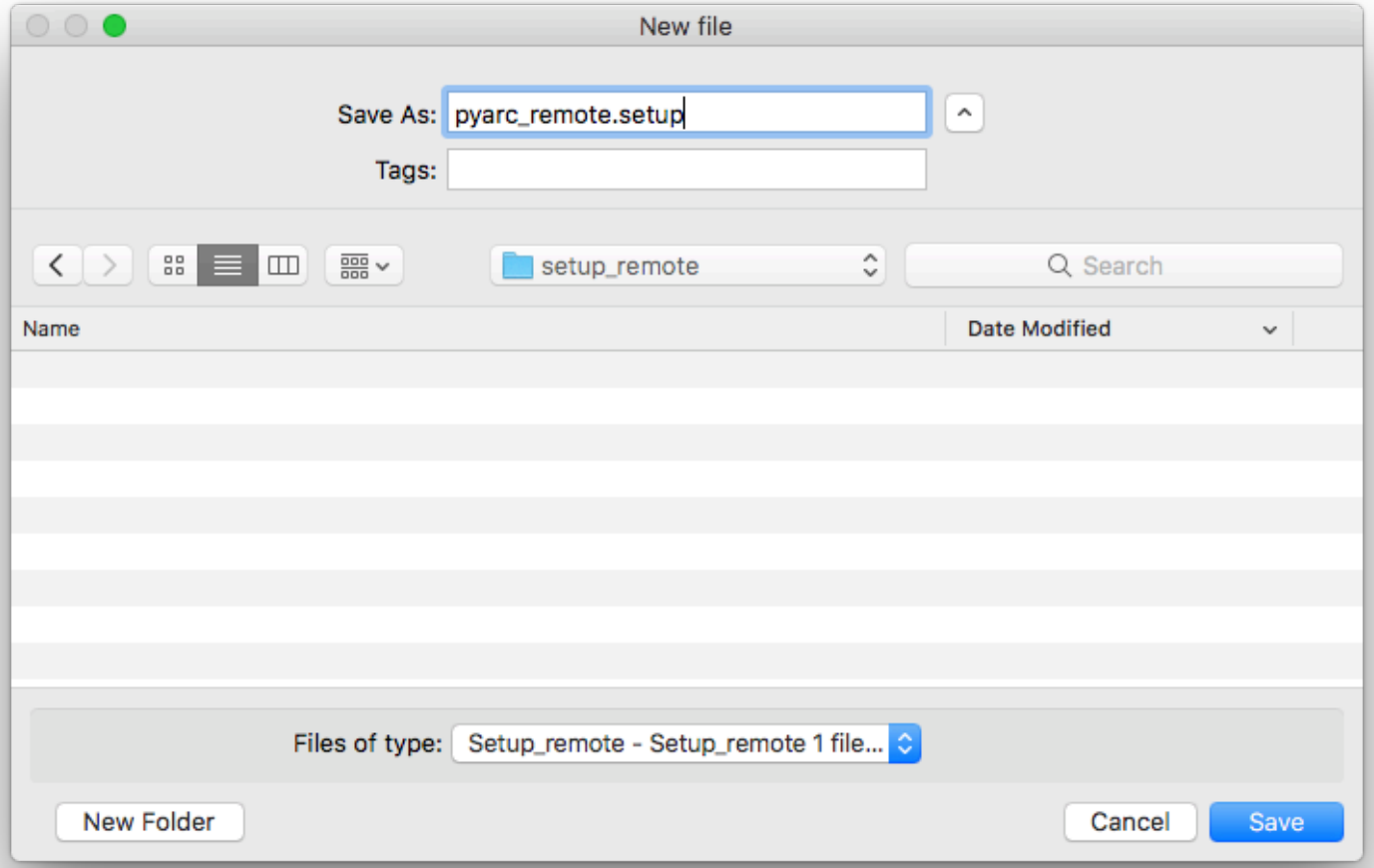


OAK RIDGE National Laboratory

Create the remote PyARC Setup file

- This connects the client/local NEAMS Workbench to the remote NEAMS Workbench and PyARC installation

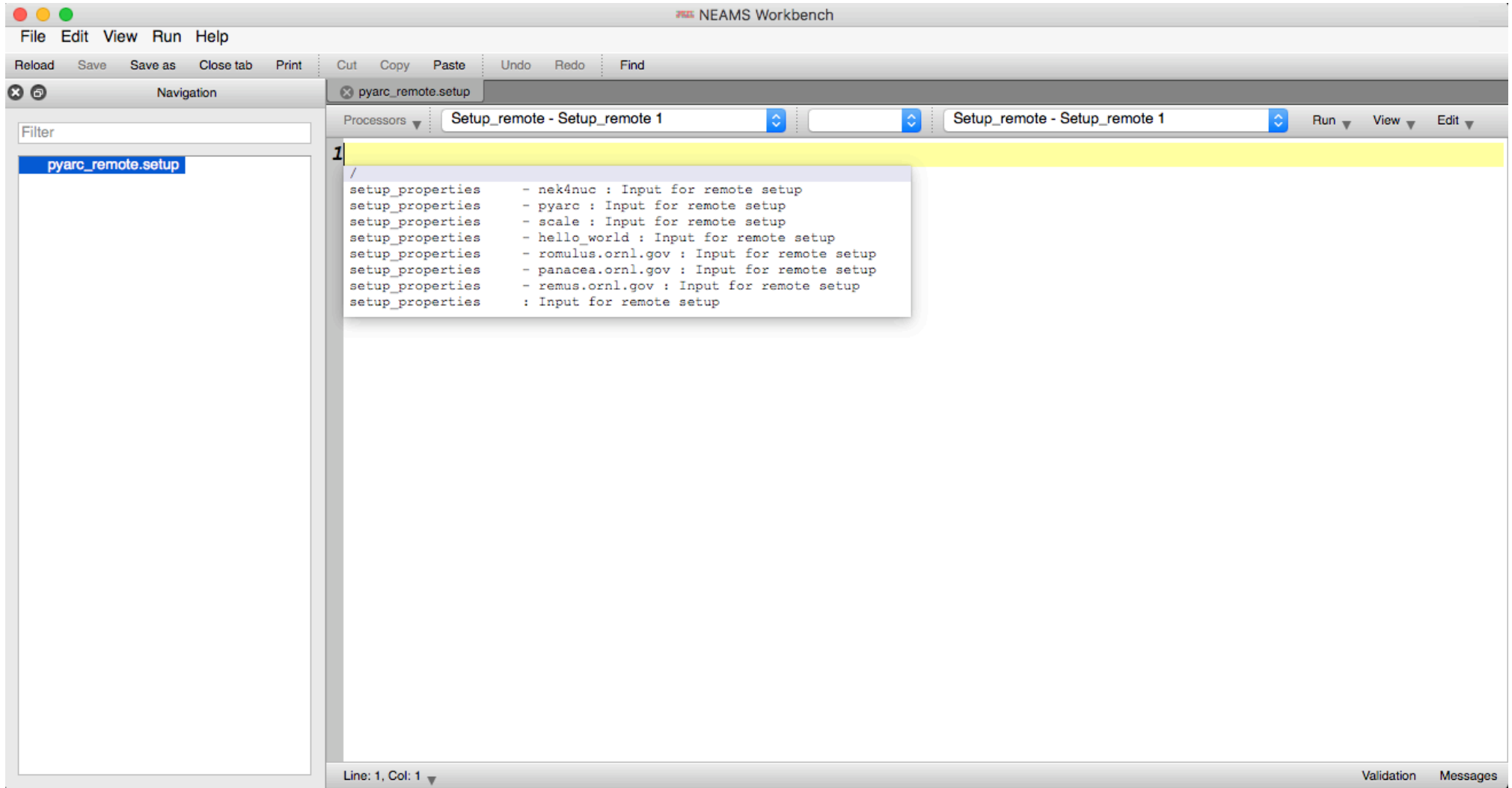
1. Click **File>New file…**
2. Provide file name of **pyarc_remote.setup** and select **Setup_remote** as the file type.
3. Click **Save**



Oak Ridge National Laboratory

Create the remote PyARC Setup file

- Click inside the input editor window and use **CTRL+SPACE** key sequence to display autocompletion options.
- Select PyARC variant



OAK RIDGE National Laboratory

Specify the remote PyARC Setup file parameters

1. Click **Edit>Find** to display the Find widget
2. Replace **jdoe** with your user id

NEAMS Workbench

File Edit View Run Help

Reload Save Save as Close tab Print Cut Copy Paste Undo Redo Find

Navigation

Filter

- pyarc_remote.setup
  - document
    - setup_properties
      - runtime_command
      - application_path
      - hostname
      - username
      - remote_execution_home
      - application_name
      - upload_patterns
      - download_patterns
      - version_number
      - scheduler_type
      - scheduler_specs

pyarc_remote.setup

Processors | Setup_remote - Setup_remote 1 | document | Setup_remote - Setup_remote 1

```
1  setup_properties {
2
3      runtime_command = "/home/jdoe/workbench/rte/entry.sh /home/jdoe/workbench/rte/arc.py"
4      application_path = "/home/jdoe/workbench/PyARC/PyARC.py"
5      hostname = "hpc.example.gov"
6      username = "jdoe"
7      remote_execution_home = "/home/jdoe"
8      application_name = "pyarc_remote" % include the client and hostname to prevent collisions from other installat
9      upload_patterns = ["*.decay_chain.son" "*.isotxs" "*.lumped.son"] % using Python's glob syntax https://docs.py
10     download_patterns = ["${BASENAME}*"] % using Python's glob syntax https://docs.python.org/2/library/glob.html
11
12     version_number = "1.0.0"
13
14     scheduler_type = pbs
15     scheduler_specs {
16         submit_path  = "/usr/local/bin/qsub"
17         status_path  = "/usr/local/bin/qstat"
18         delete_path  = "/usr/local/bin/qdel"
19         hold_path    = "/usr/local/bin/qhold"
20         release_path = "/usr/local/bin/qrls"
21     }
```

Find: jdoe Case Sensitive

Replace with: Replace | Replace & find | Replace all

Line: 1, Col: 1 /setup_properties/decl - Input for remote setup Validation Messages

Specify the remote PyARC Setup file parameters

1. Replace **workbench** with your server-side NEAMS Workbench installation

NEAMS Workbench

File Edit View Run Help

Reload Save Save as Close tab Print Cut Copy Paste Undo Redo Find

Navigation

Filter

- pyarc_remote.setup
  - document
    - setup_properties
      - runtime_command
      - application_path
      - hostname
      - username
      - remote_execution_home
      - application_name
      - upload_patterns
      - download_patterns
      - version_number
      - scheduler_type
      - scheduler_specs

pyarc_remote.setup

Processors | Setup_remote - Setup_remote 1 | document | Setup_remote - Setup_remote 1

```
1  setup_properties {
2
3      runtime_command = "/home/jdoe/workbench/rte/entry.sh /home/jdoe/workbench/rte/arc.py"
4      application_path = "/home/jdoe/workbench/PyARC/PyARC.py"
5      hostname = "hpc.example.gov"
6      username = "jdoe"
7      remote_execution_home = "/home/jdoe"
8      application_name = "pyarc_remote" % include the client and hostname to prevent collisions from other installat
9      upload_patterns = ["*.decay_chain.son" "*.isotxs" "*.lumped.son"] % using Python's glob syntax https://docs.py
10     download_patterns = ["${BASENAME}*"] % using Python's glob syntax https://docs.python.org/2/library/glob.html
11
12     version_number = "1.0.0"
13
14     scheduler_type = pbs
15     scheduler_specs {
16         submit_path  = "/usr/local/bin/qsub"
17         status_path  = "/usr/local/bin/qstat"
18         delete_path  = "/usr/local/bin/qdel"
19         hold_path    = "/usr/local/bin/qhold"
20         release_path = "/usr/local/bin/qrls"
21     }
```

Find: workbench | Case Sensitive

Replace with: | Replace | Replace & find | Replace all

Line: 3, Col: 5 /setup_properties/runtime_command/decl - Executable command for the runtime on the host Validation Messages

Oak National

Specify the remote PyARC Setup file parameters

1. Replace **workbench** with your server-side NEAMS Workbench installation
2. Update the **hostname** to be the server name on which the NEAMS Workbench and application are installed
3. Review and update as needed the **scheduler_specs**
   - Important for indicating scheduler queues or job runtime or memory

OAK RIDGE National Laboratory

Run the remote PyARC Setup file

- Click the **run** button to execute the PyARC setup file

NEAMS Workbench

File Edit View Run Help

Reload Save Save as Close Print Cut Copy Paste Undo Redo Find

Navigation

Filter

- pyarc_remote.setup
  - document
    - setup_properties
      - runtime_command
      - application_path
      - hostname
      - username
      - remote_execution_home
      - application_name
      - upload_patterns
      - download_patterns
      - version_number
      - scheduler_type
      - scheduler_specs

pyarc_remote.setup

Processors | Setup_remote - Setup_remote 1 | document | Setup_remote - Setup_remote 1 | Run

```
1 setup_properties {
2
3     runtime_command = "/home/raq/workbench/rte/entry.sh /home/raq/workbench/rte/arc.py"
4     application_path = "/home/raq/workbench/PyARC/PyARC.py"
5     hostname = "hpc.example.gov"
```

Fri Nov 15 14:30:45 2019

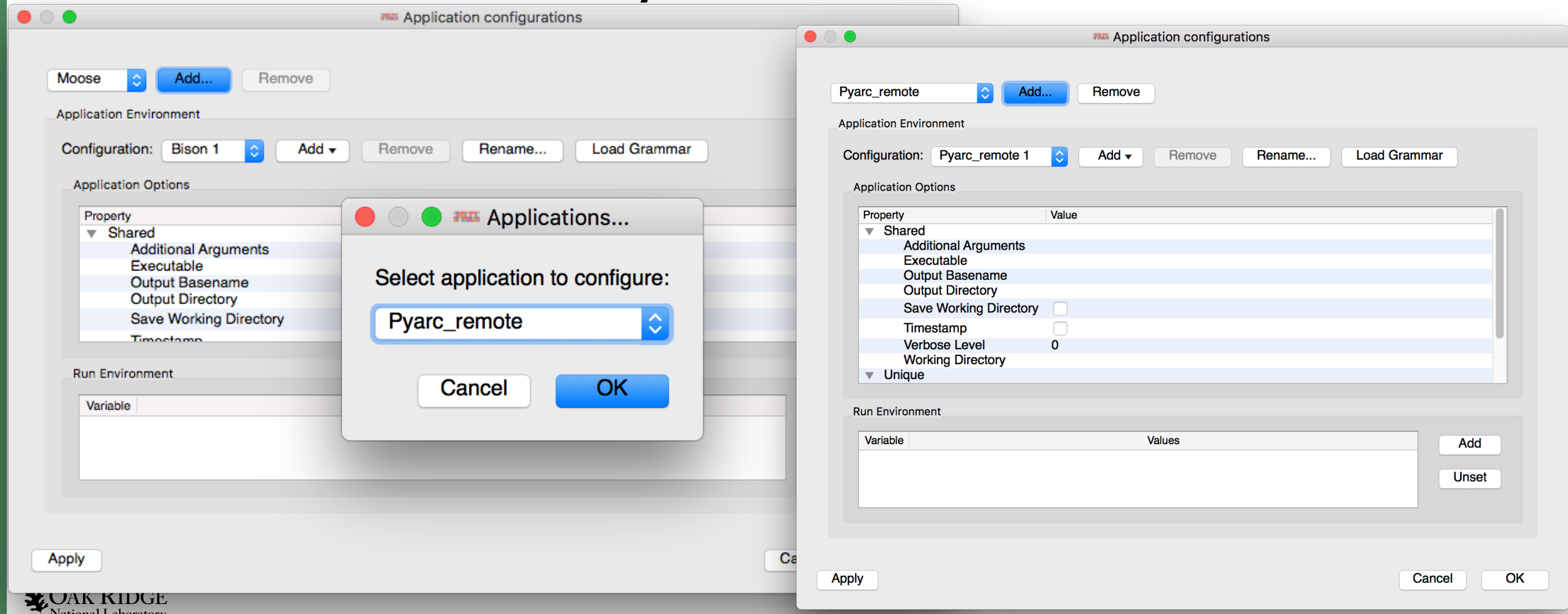
```
24 Completed writing server configuration file: /projects/NEAMS/RELEASES/downloads/Workbench-Darwin/rte/cirrus/packages/deplo
25
26 Completed writing server Python script: /projects/NEAMS/RELEASES/downloads/Workbench-Darwin/rte/cirrus/packages/deployed/p
27
28 Communicating via native ssh: /usr/bin/ssh
29 No server specified.  Unable to verify communication.
30
31 Attempting to upload to hpc.example.gov
32
33 Communicating via native ssh: /usr/bin/ssh
34 Verifying server credentials...
35
36 Verified server credentials are valid.
37
38 Completed uploading to hpc.example.gov:/home/raq/workbench/rte/cirrus/packages/
39
40 Verifying expected transmission path on hpc.example.gov...
41
42 Verified expected transmission path on hpc.example.gov: /home/raq/.cirrus/warroom7_hpc.example.gov/pyarc_remote_server.py
43
44 Completed writing the class file at /projects/NEAMS/RELEASES/downloads/Workbench-Darwin/rte/pyarc_remote.py
45 Loading pyarc_remote module...
46 Attempting to download the options file
47 Executing remotely '/home/raq/workbench/rte/entry.sh /home/raq/workbench/rte/arc.py -json'
48
49 Completed downloading the options file at /projects/NEAMS/RELEASES/downloads/Workbench-Darwin/rte/remote_resources/pyarcRe
50
51 Process finished with 0 return code; ran in 19 secs, finished at Fri Nov 15 14:31:04 2019
```

Line: 11, Col: 1 /setup_properties - Input for remote setup Validation Messages

OAK RID National Labor

Configure the **PyARC_remote** application

- Click **File>Configurations…**
- Click **Add…** and select **Pyarc_remote** and click **OK**

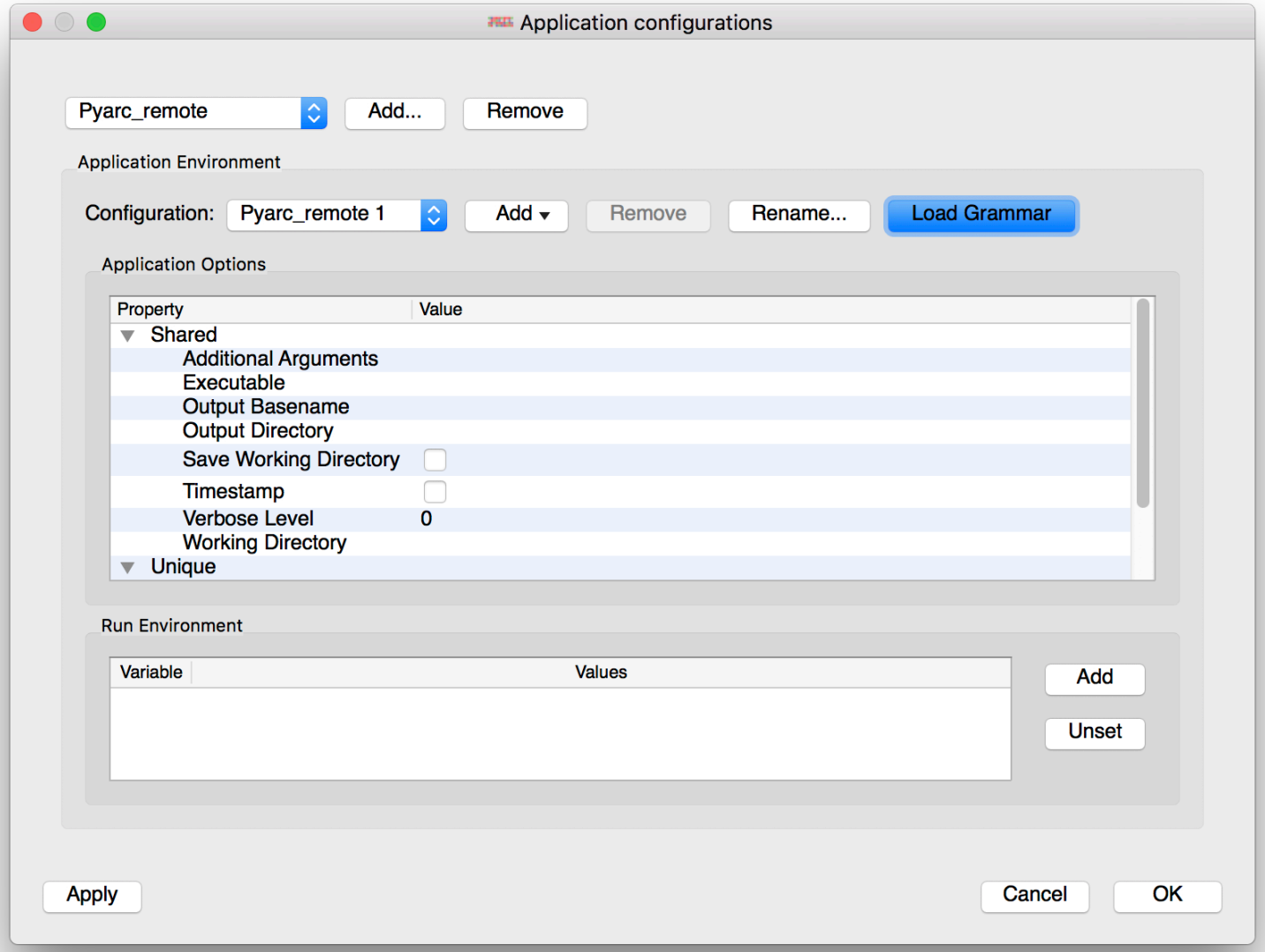


OAK RIDGE National Laboratory

Configure the **PyARC_remote** application

- Click **Load Grammar**, **Apply**, and **OK.** This enables PyARC input creation.

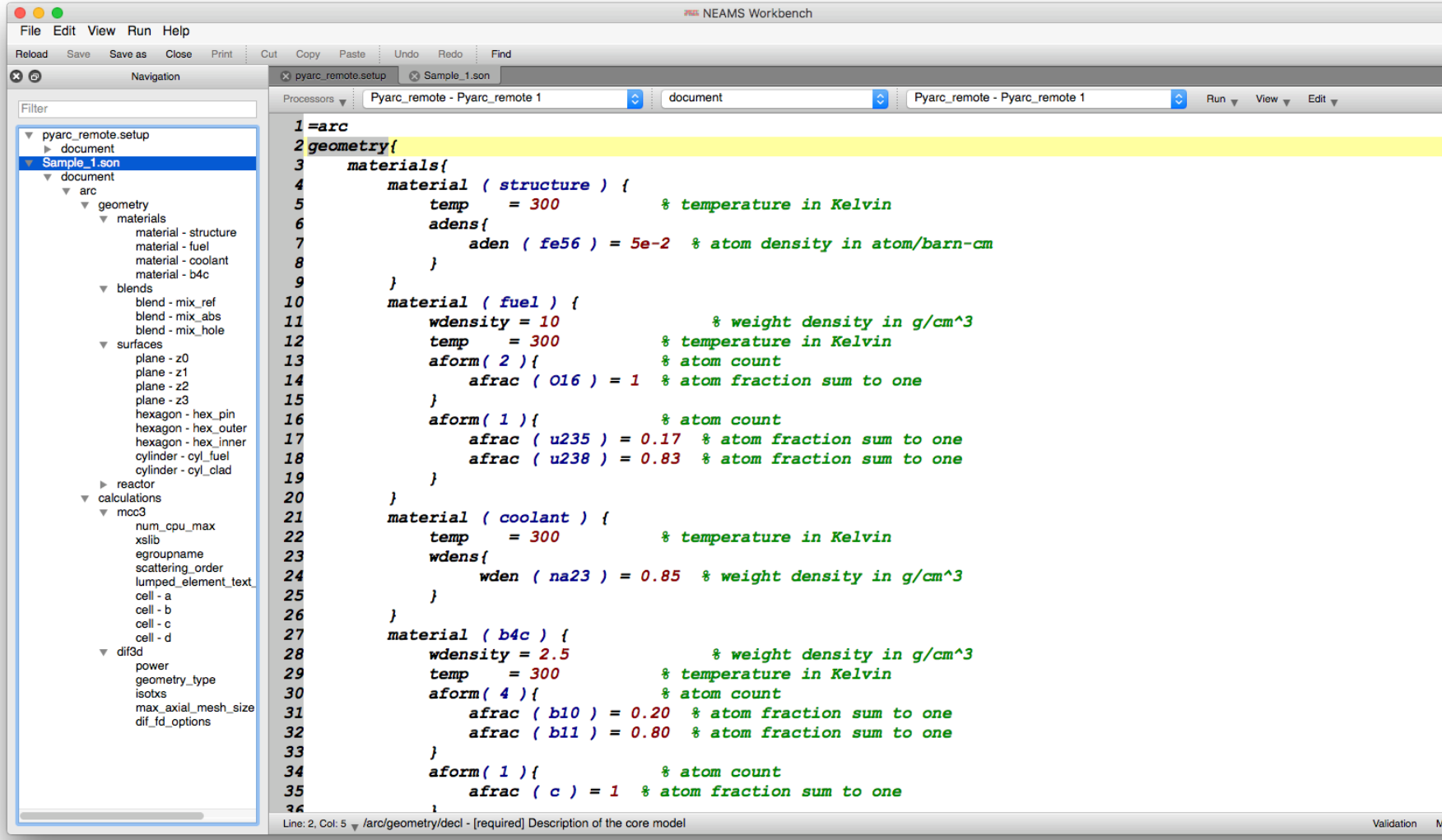
- Note: Load grammar may pause for a moment while Workbench retrieves the grammar files from the server.



OAK RIDGE National Laboratory

Open the PyARC Sample_1 Input

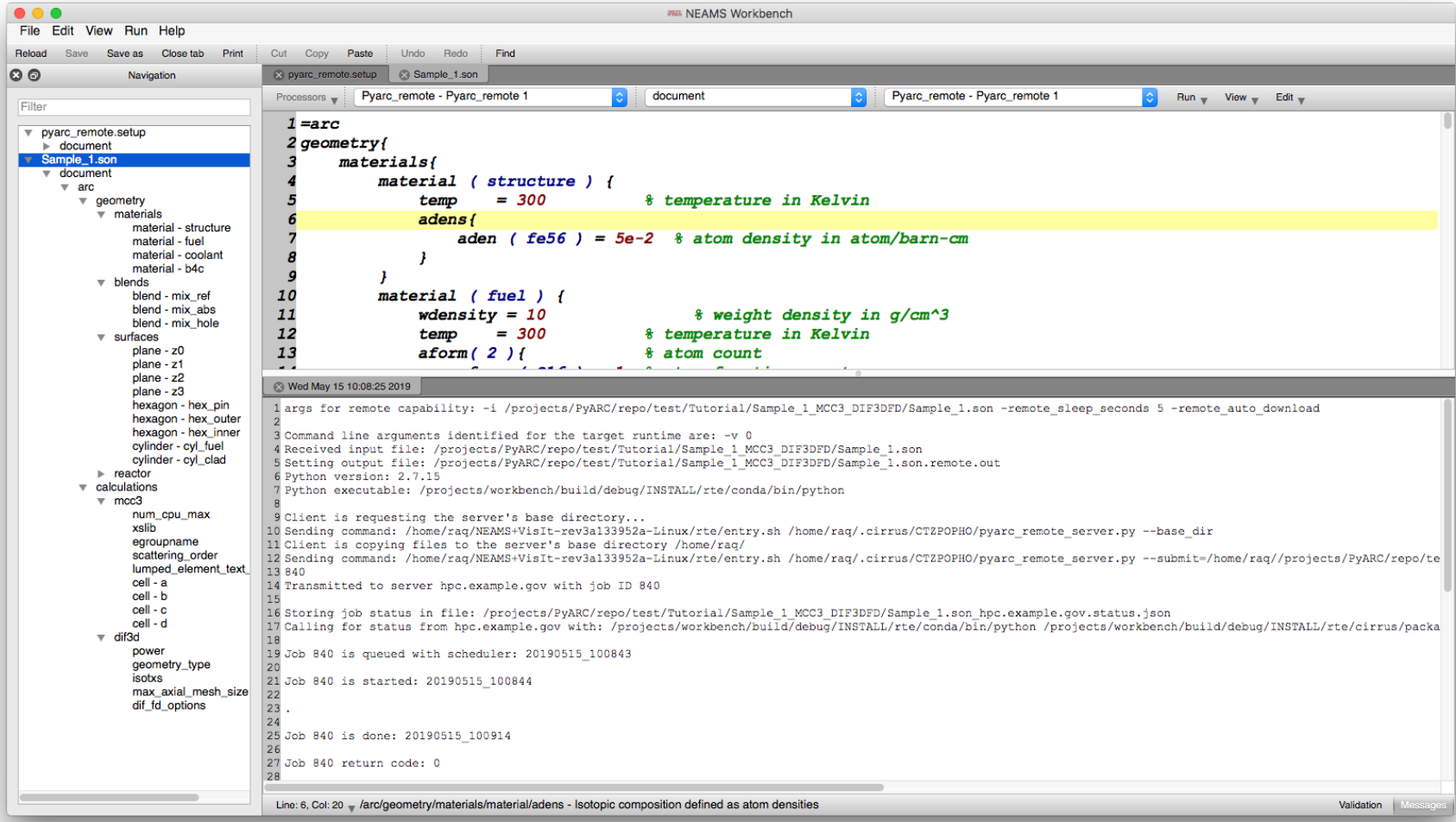
- Click **File>Open…** and navigate to and open the Workbench/**PyARC/tutorial/Sample_1_MCC3_DIF3DFD /Sample_1.son**
- select the **Pyarc_remote** run configuration



OAK RIDGE National Laboratory

Run the PyARC Sample_1 Input

- Click the **Run** button to launch the job on the remote system
- Select the **Messages** button to display the messages panel
- Wait for the job to complete
- Ensure the output states **Successfully downloaded Sample_1.dif3d.vtk**



OAK RIDGE National Laboratory

Run Configurations

- The Run Configuration is what communicates the application's grammar to the NEAMS Workbench
  - The grammar is the definition of how to interpret the input syntax, data, etc.
- Note: If you open an application input and there is no grammar available (no input navigation tree), check that your application's run configuration is setup.

OAK RIDGE National Laboratory

Questions?

- Email questions and issues to nwb-help@ornl.gov

OAK RIDGE National Laboratory