

NEAMS Workbench Dakota 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

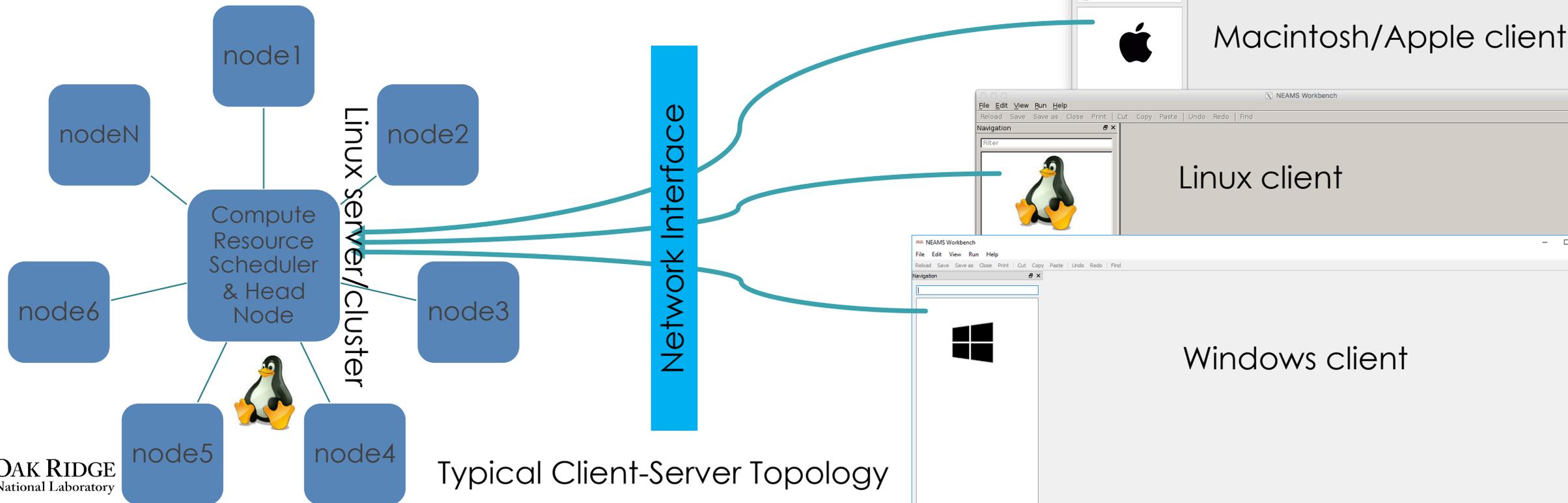
Assumptions

- 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 Dakota code has been installed
 - Dakota is freely available from <https://dakota.sandia.gov/download.html>
 - Download and unzip/tar.gz platform install

Need assistance? Email nwb-help@ornl.gov

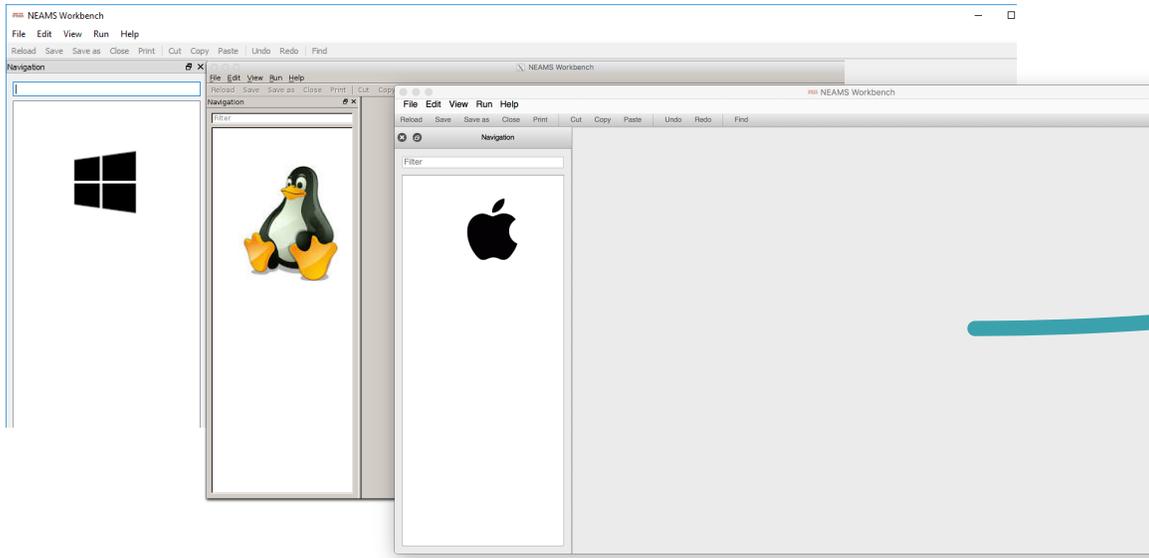
Client vs. Server definition

- ‘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

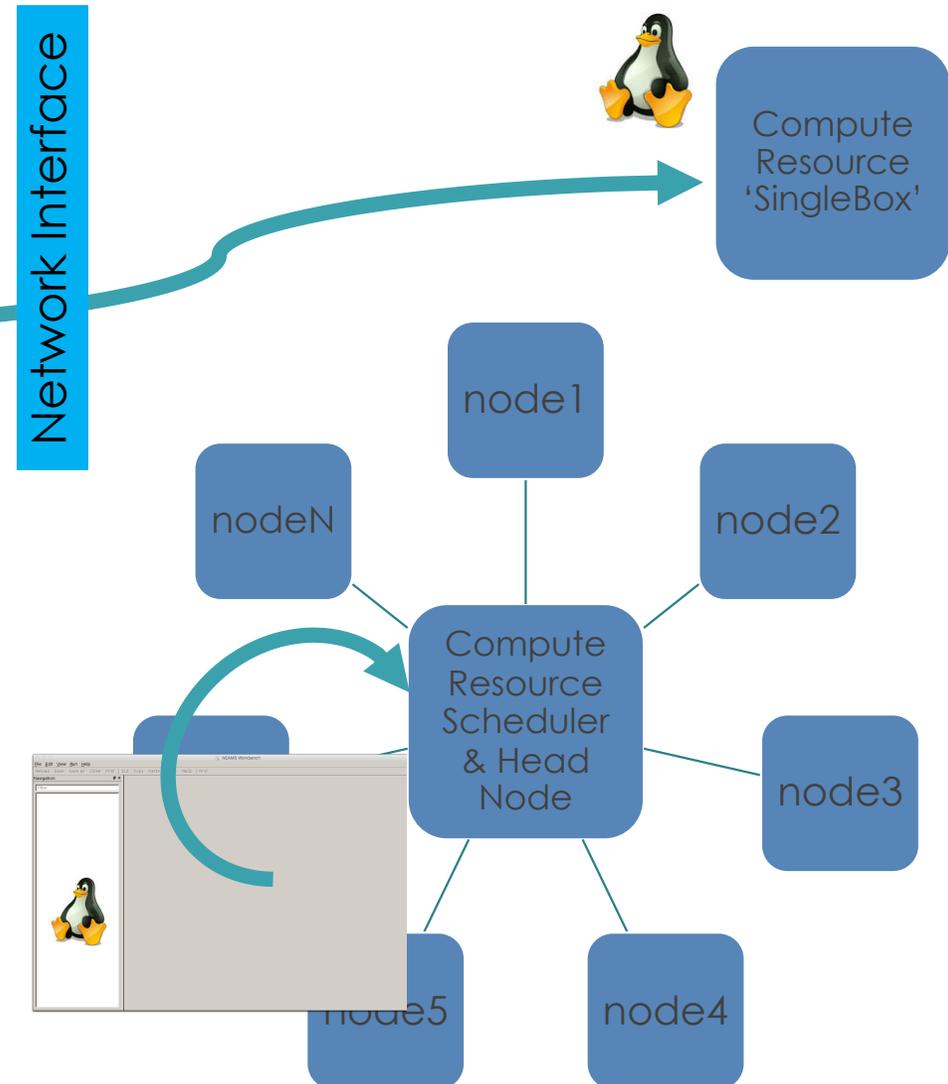


Additional configurations

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

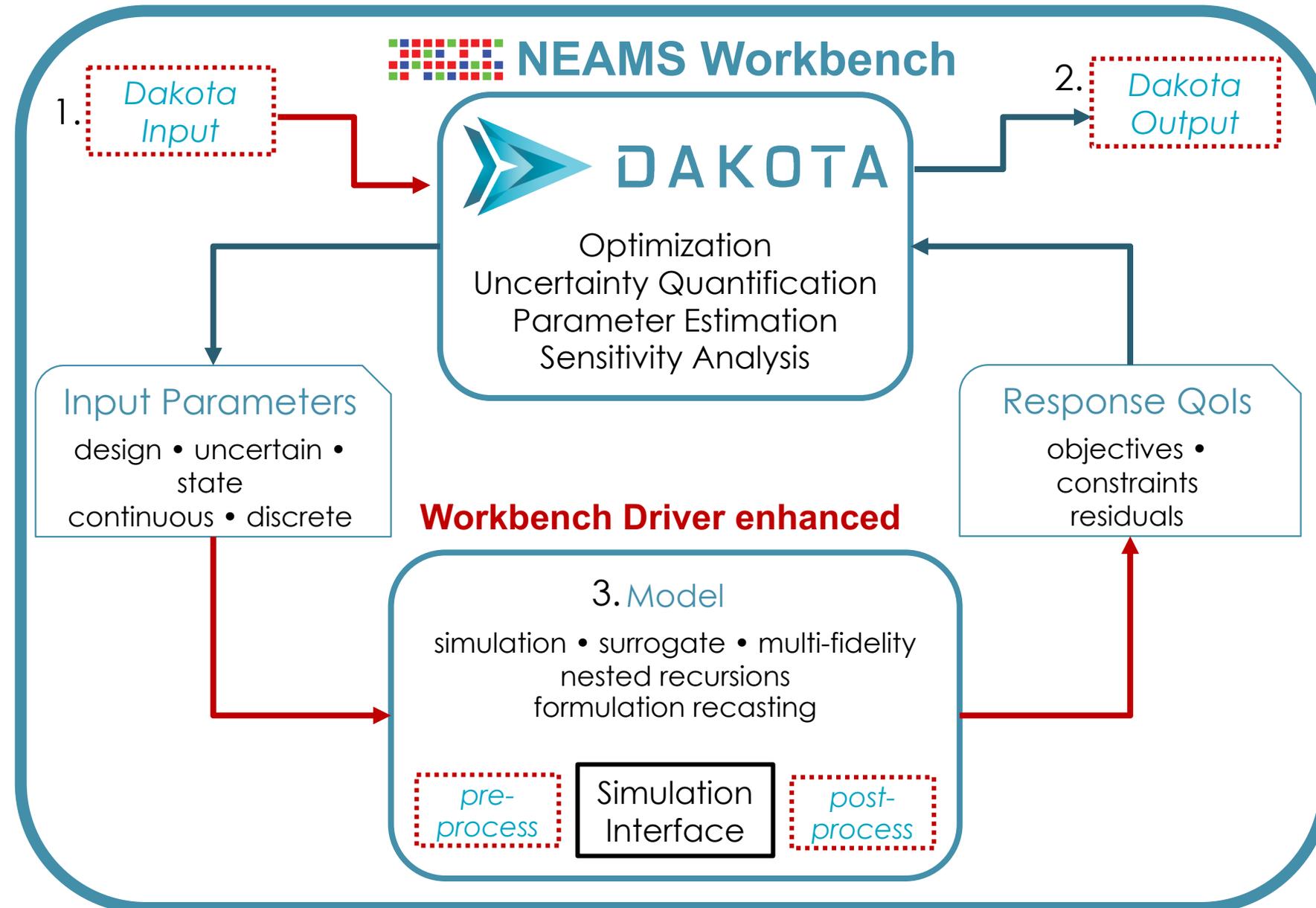


- Localhost – client is the cluster head node with scheduler from which jobs are submitted locally to the scheduler



Dakota within the NEAMS Workbench

1. Dakota Input:
 - Autocompletion
 - verification
2. Dakota Output
 - results plotting
3. Model:
 - Workbench Driver
 - Preprocessing
 - Template editing
 - Postprocessing



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>

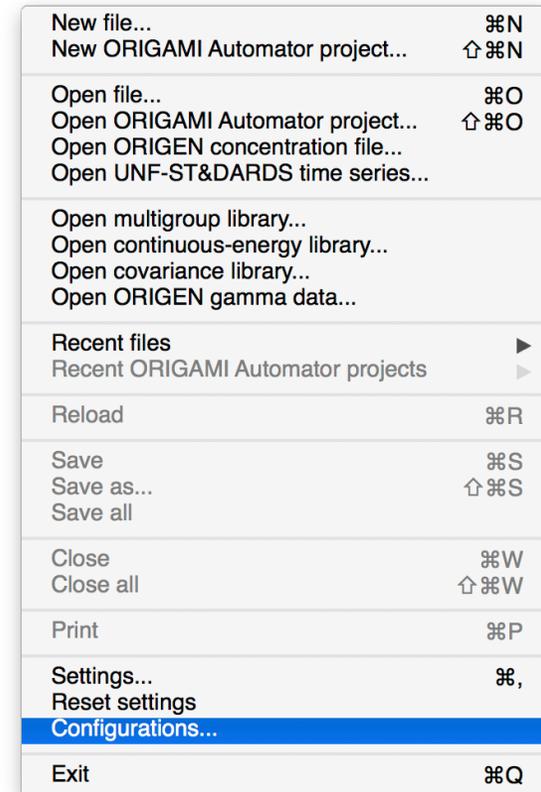
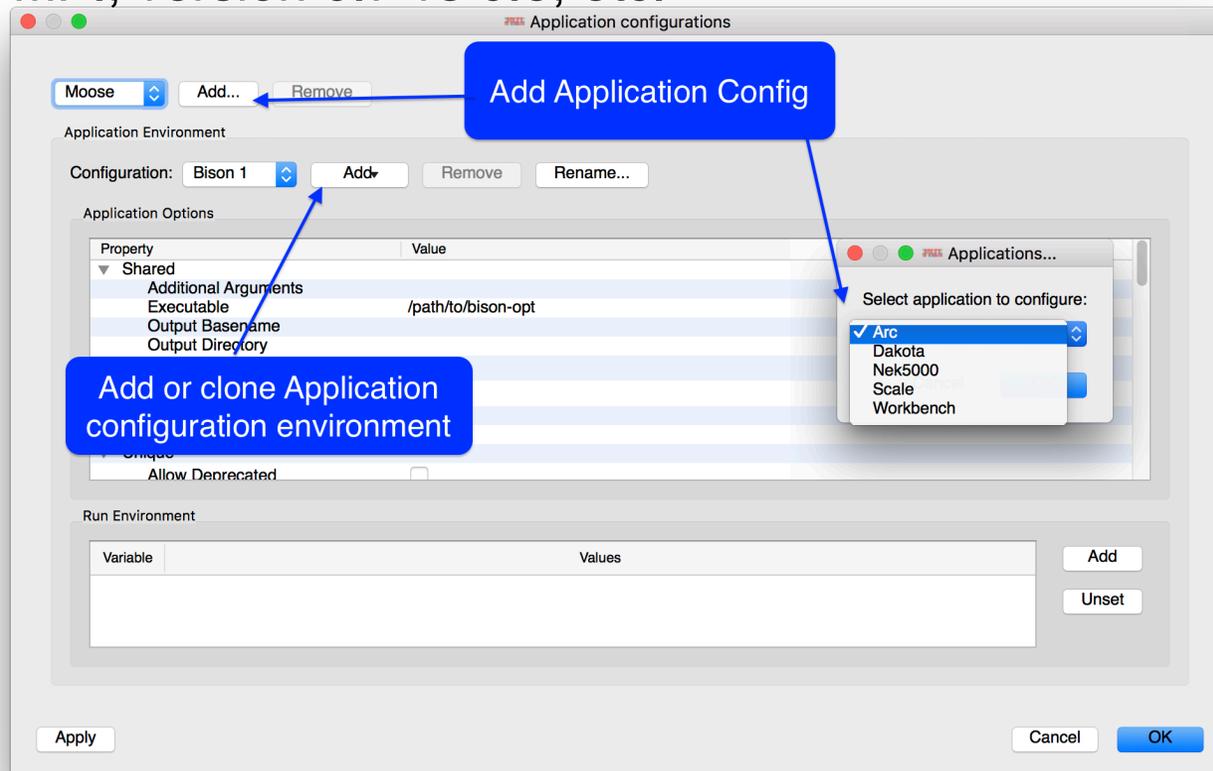
NOTE: be sure to use the same client and server versions. Server installations can be shared between users as long as the administrating user executes the `Workbench-Linux/rte/entry.sh` to configure the necessary Python environment

Local and Remote Application Setup Overview

- Local application setup requires installation of the Dakota code 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

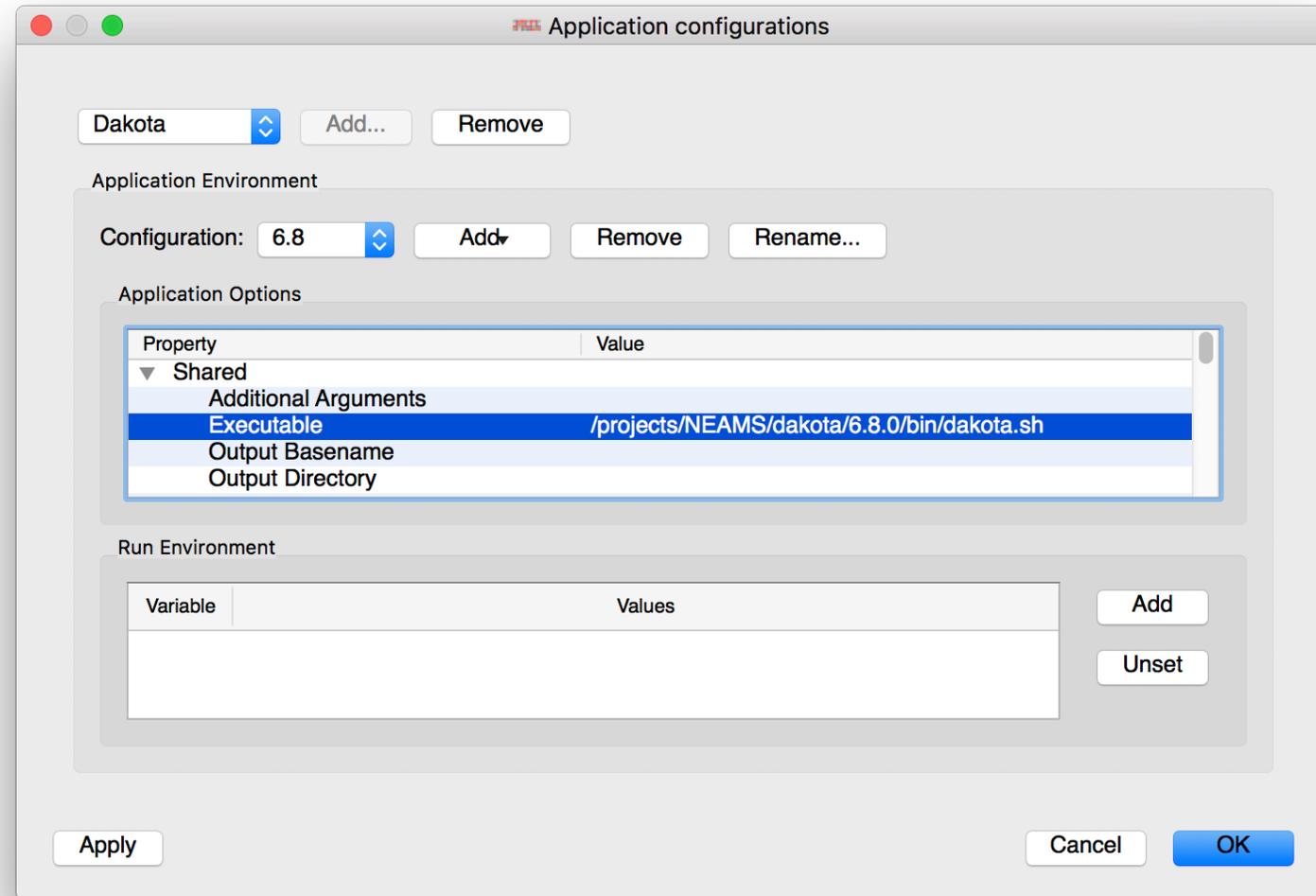
Run Configurations - Overview

- 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.



Local Configuration of Dakota UQ and Optimization*

- Open the Application Configuration dialog via **File>Configurations**
- Select **Add... Dakota**
- Update the **Executable** property's value to be the absolute path to **dakota.sh** or **dakota.bat**
- A pause can be observed as the NEAMS Workbench requests/generates the Dakota input definition



Remote Run Configuration of Dakota 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 Dakota remote run configuration.
 - d. Configure the Dakota_remote
 - e. Open any Dakota input and select the Dakota remote run configuration to activate input validation, autocompletion, syntax highlighting and job launch capabilities
 - f. Edit input as needed
 - g. Use the Dakota remote run configuration to launch the Dakota input from your client to the server
 - h. Wait for job to finish
 - i. Open files associated with the input file (*.out, etc.)

Passwordless SSH Setup

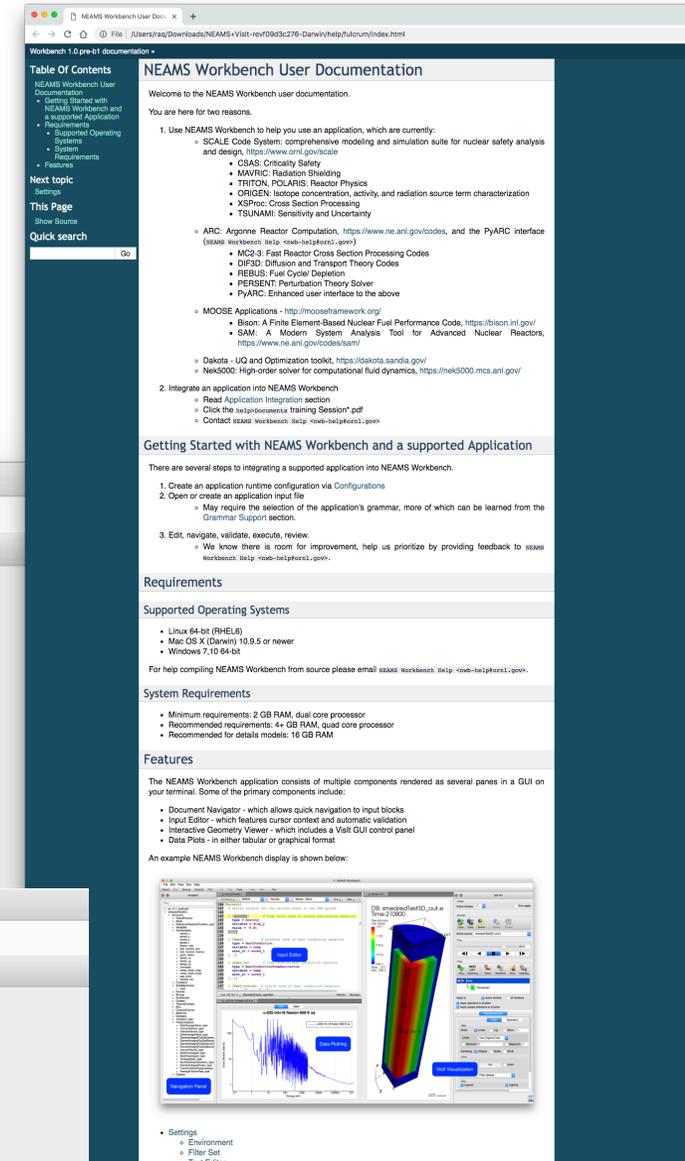
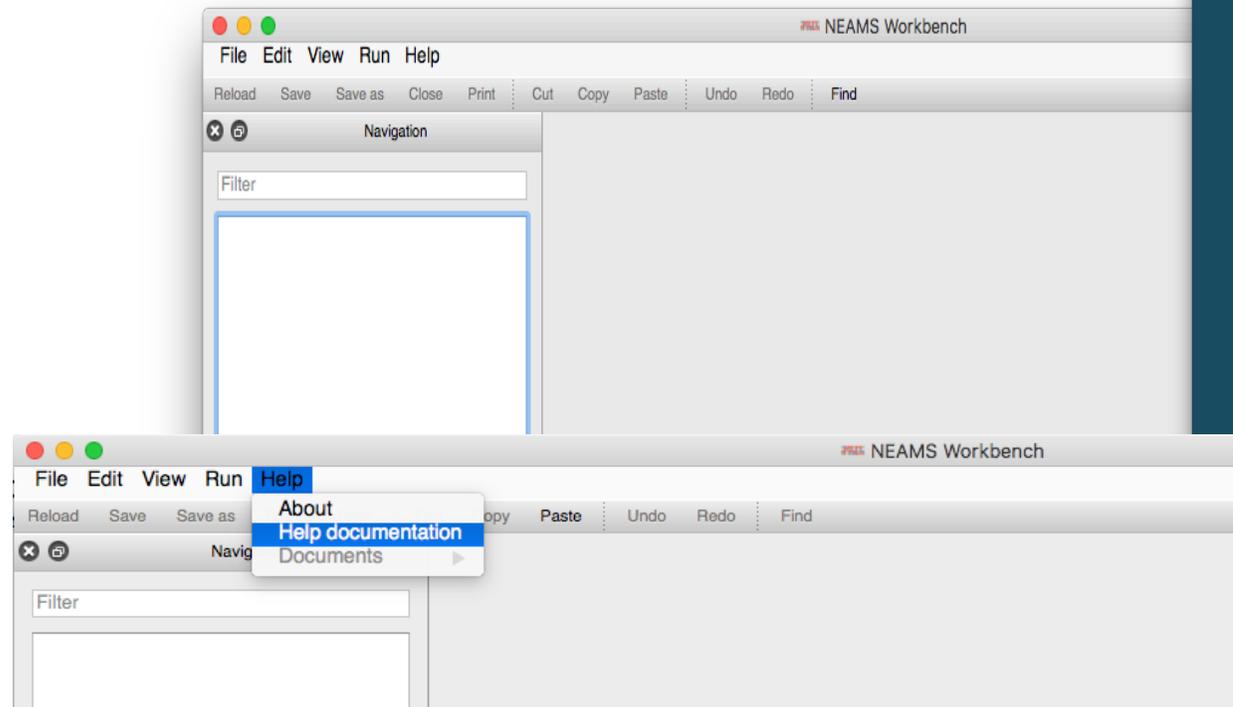
- Required for remote run configurations
- Appends client machine's authorized public key to the user's list of authorized keys residing on the server. The following statement is sufficient to do so on **Linux** or **Mac** clients:

```
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"

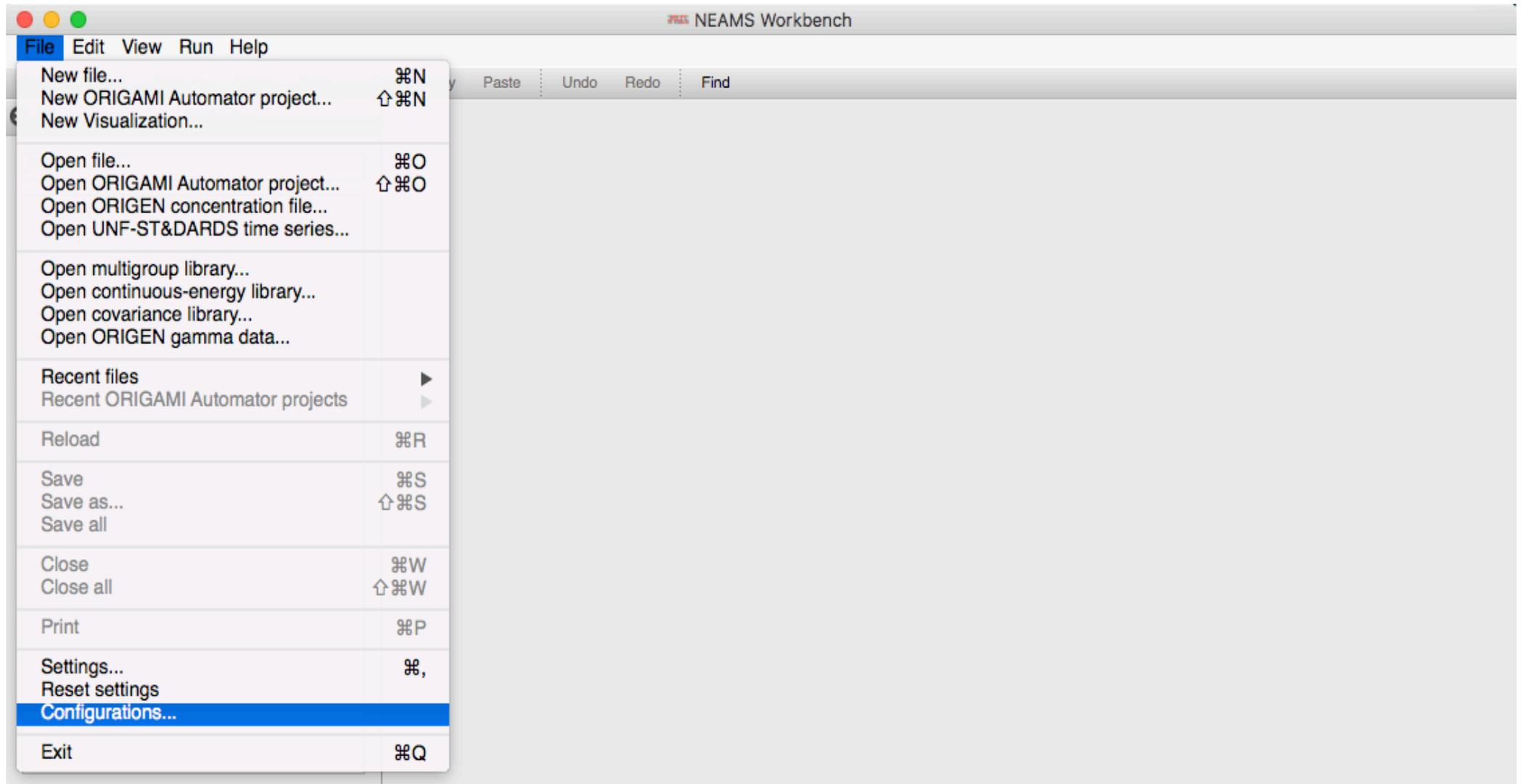
Start the client install of the NEAMS Workbench

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



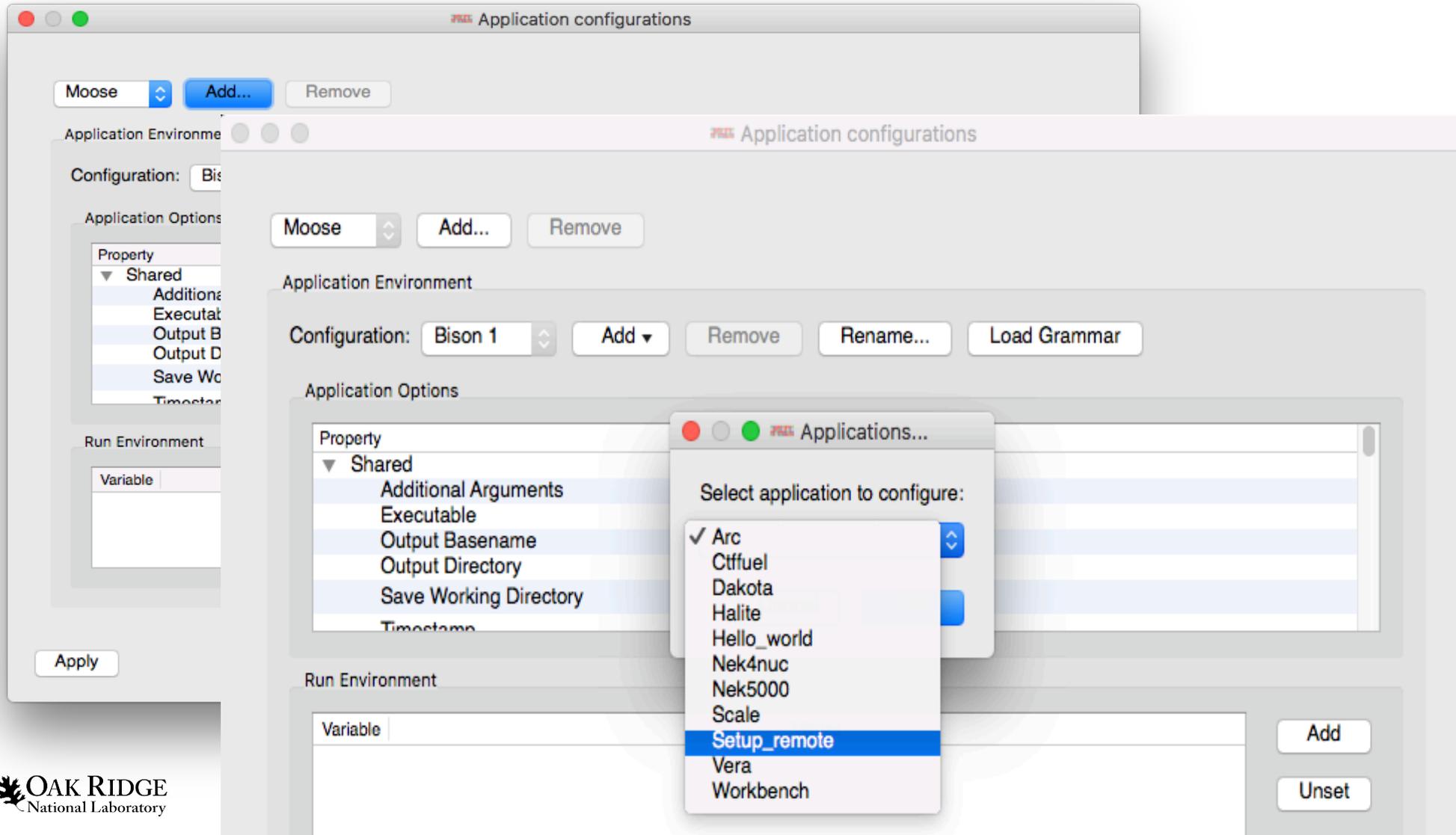
Configure the Setup Remote Application

- Click **File>Configurations...**



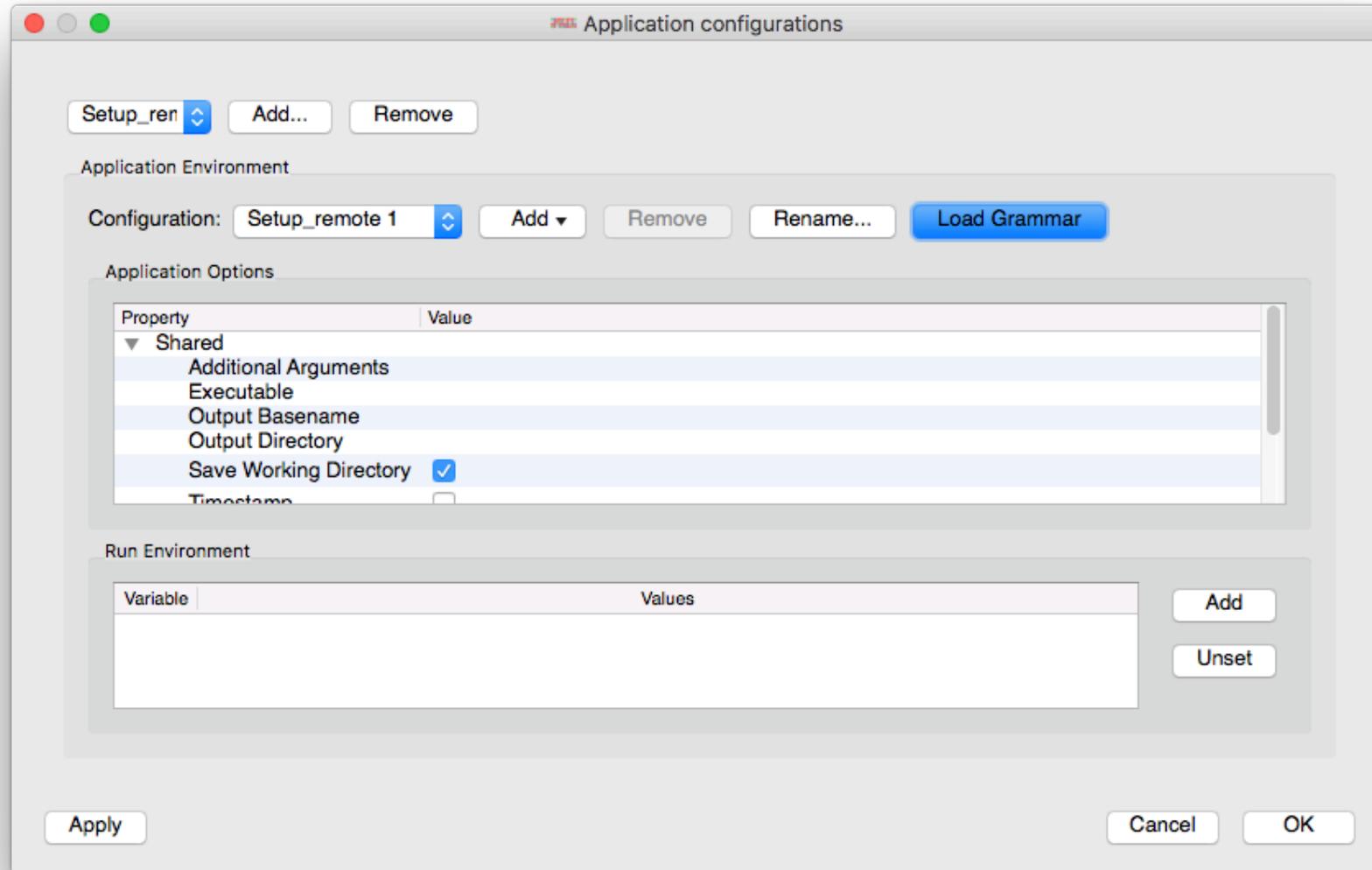
Configure the Setup Remote Application

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



Configure the Setup Remote Application

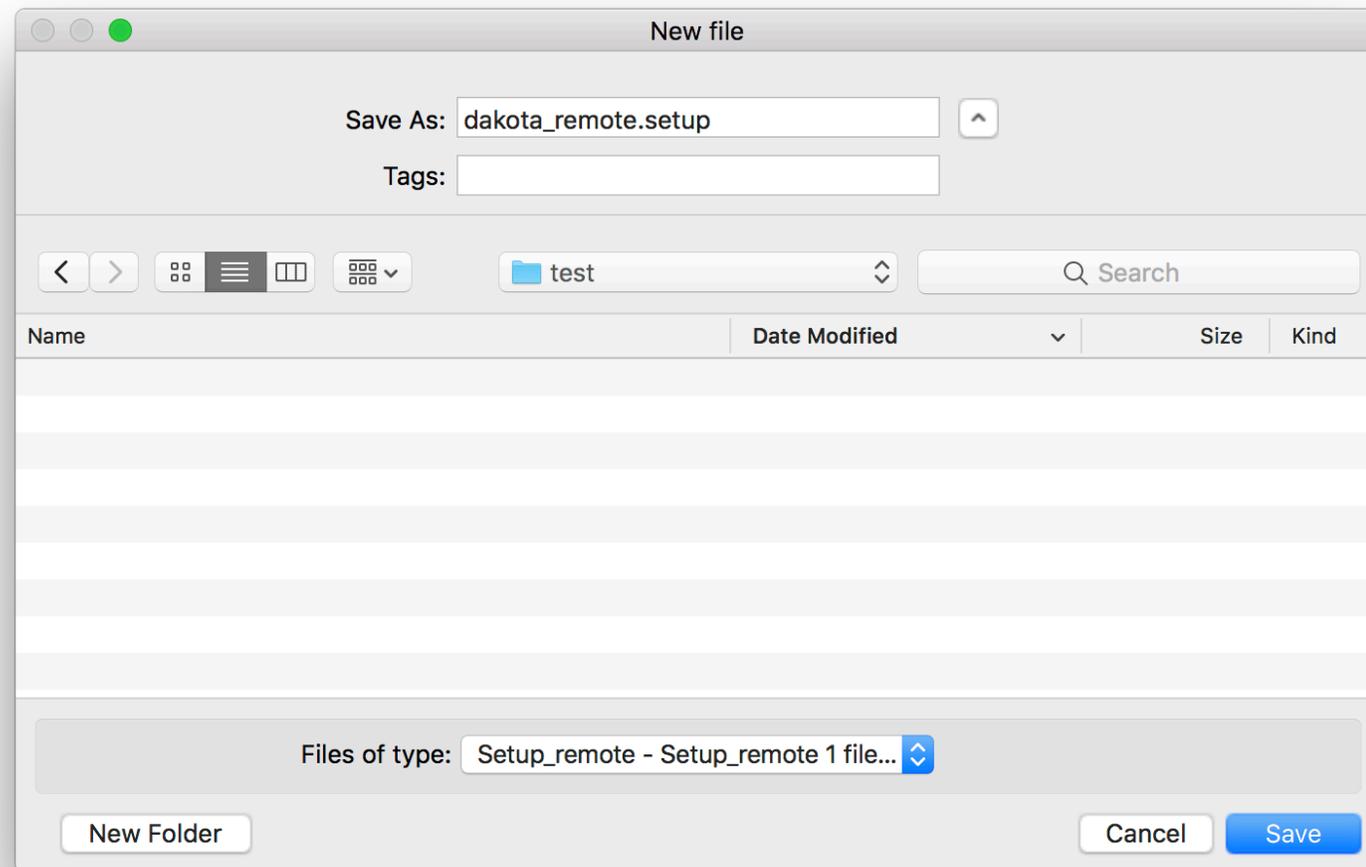
- Click **Load Grammar**, **Apply**, and **OK**. This enables SetupRemote input creation.



Create the remote Dakota Setup file

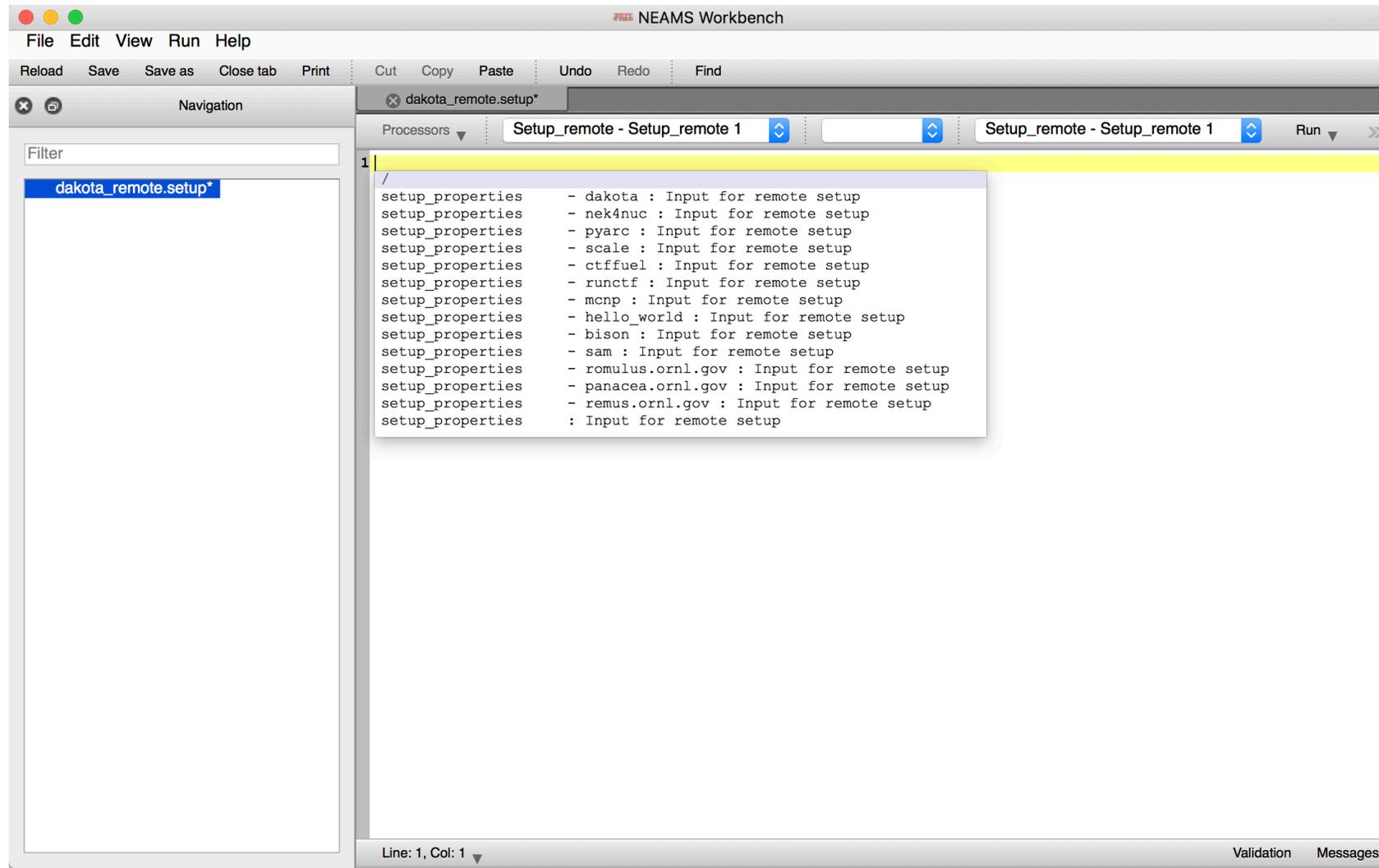
- This connects the client/local NEAMS Workbench to the remote NEAMS Workbench and Dakota installation

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



Create the remote Dakota Setup file

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



Specify the remote Dakota Setup file parameters

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

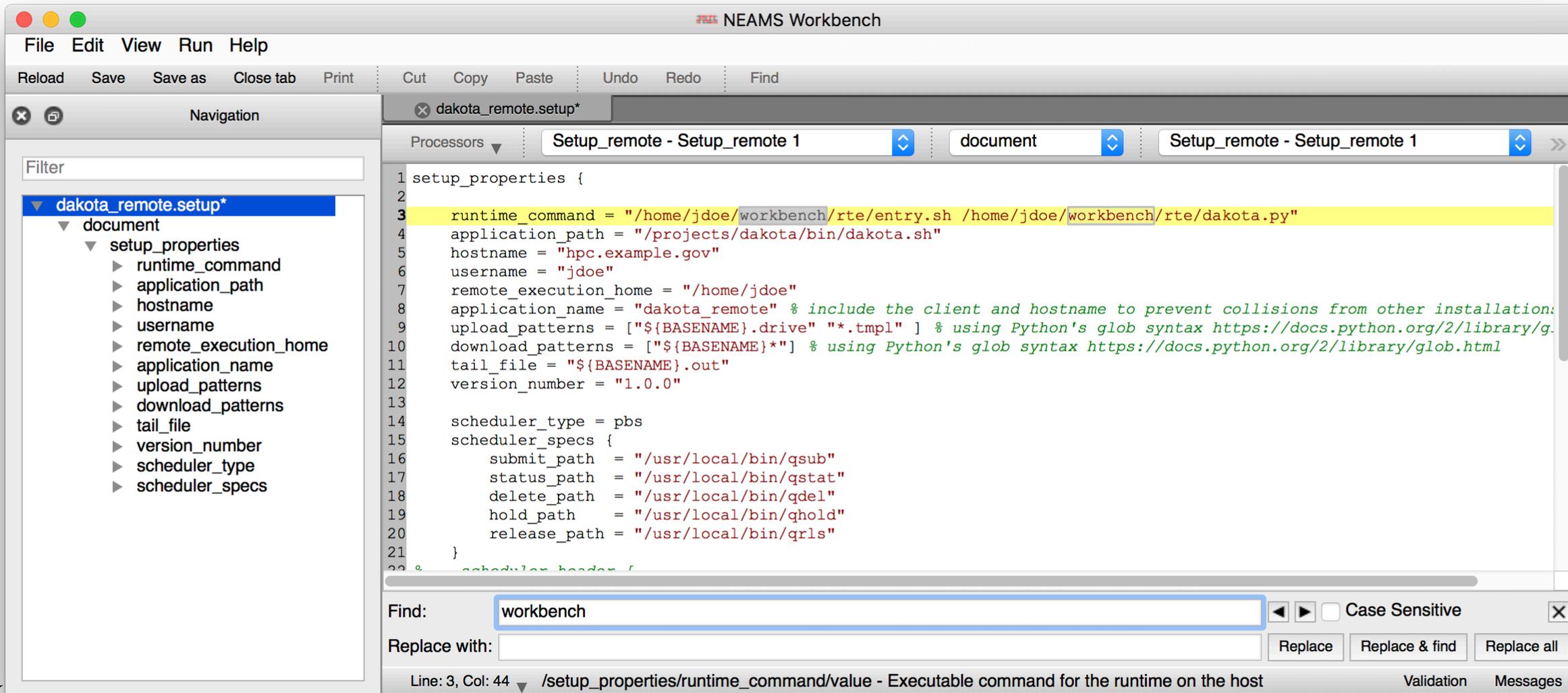
The screenshot shows the NEAMS Workbench interface. The main window displays the file `dakota_remote.setup` with the following content:

```
1 setup_properties {
2
3   runtime_command = "/home/jdoe/workbench/rte/entry.sh /home/jdoe/workbench/rte/dakota.py"
4   application_path = "/projects/dakota/bin/dakota.sh"
5   hostname = "hpc.example.gov"
6   username = "jd"oe"
7   remote_execution_home = "/home/jdoe"
8   application_name = "dakota_remote" % include the client and hostname to prevent collisions from other installations
9   upload_patterns = ["${BASENAME}.drive" "*.templ" ] % using Python's glob syntax https://docs.python.org/2/library/g
10  download_patterns = ["${BASENAME}*" ] % using Python's glob syntax https://docs.python.org/2/library/glob.html
11  tail_file = "${BASENAME}.out"
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  }
22 }
```

The Find widget is open at the bottom of the window, showing the search term `jd`oe and the Replace with field. The status bar at the bottom indicates the cursor is at Line 8, Col 77.

Specify the remote Dakota Setup file parameters

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



The screenshot shows the NEAMS Workbench application interface. The main window displays the configuration file 'dakota_remote.setup*' with the following content:

```
1 setup_properties {
2
3 runtime_command = "/home/jdoe/workbench/rte/entry.sh /home/jdoe/workbench/rte/dakota.py"
4 application_path = "/projects/dakota/bin/dakota.sh"
5 hostname = "hpc.example.gov"
6 username = "jdoe"
7 remote_execution_home = "/home/jdoe"
8 application_name = "dakota_remote" % include the client and hostname to prevent collisions from other installation
9 upload_patterns = ["${BASENAME}.drive" "*.templ" ] % using Python's glob syntax https://docs.python.org/2/library/g
10 download_patterns = ["${BASENAME}*" ] % using Python's glob syntax https://docs.python.org/2/library/glob.html
11 tail_file = "${BASENAME}.out"
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 }
22 % scheduler_header f
```

The 'runtime_command' parameter on line 3 is highlighted in yellow. A search bar at the bottom of the window shows 'workbench' in the 'Find' field, and the 'Replace with' field is empty. The 'Replace' button is visible. The status bar at the bottom indicates 'Line: 3, Col: 44' and provides a description of the current line: '/setup_properties/runtime_command/value - Executable command for the runtime on the host'.

Specify the remote Dakota 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

Run the remote Dakota Setup file

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

The screenshot displays the NEAMS Workbench interface. The top menu bar includes File, Edit, View, Run, and Help. Below the menu is a toolbar with buttons for Reload, Save, Save as, Close, Print, Cut, Copy, Paste, Undo, Redo, and Find. The main workspace is divided into three panes:

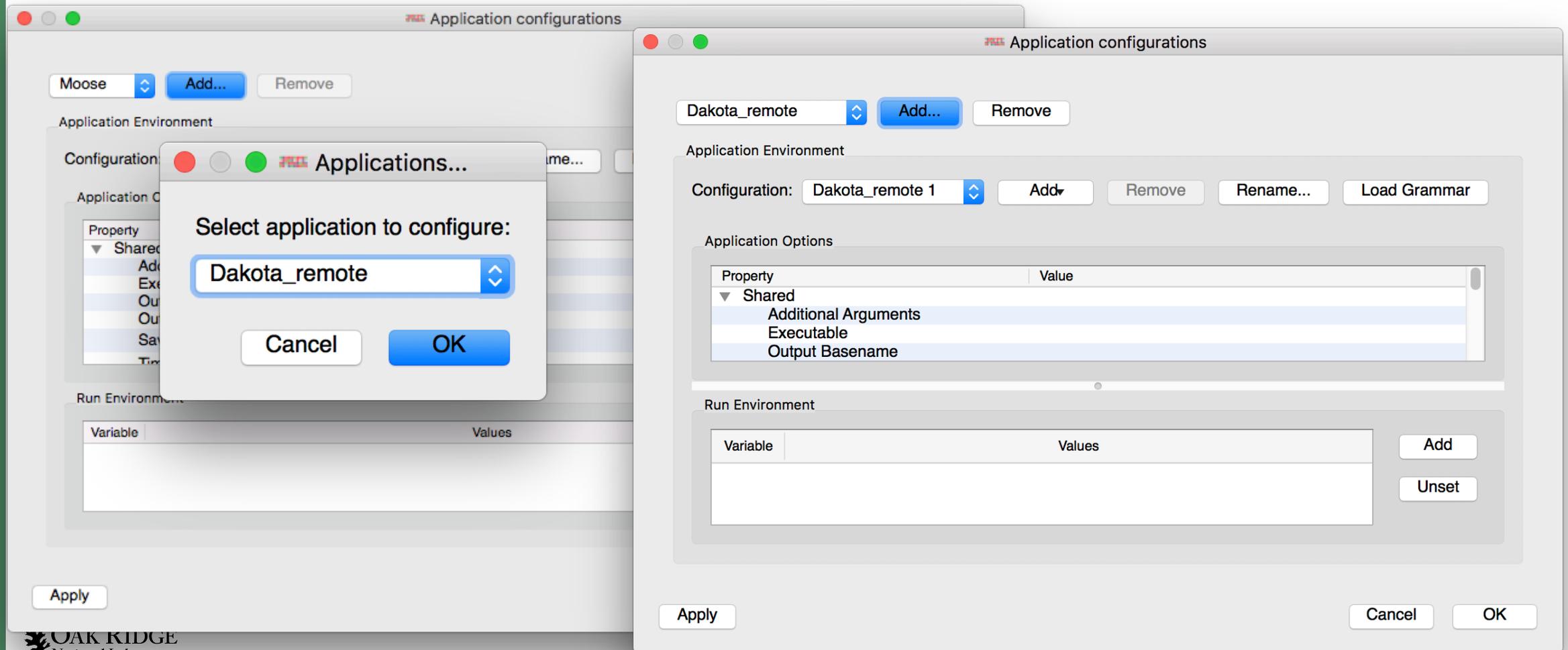
- Navigation Pane:** Shows a tree view of the project structure. The 'dakota_remote.setup' file is selected, and its properties are listed: runtime_command, application_path, hostname, username, remote_execution_home, application_name, upload_patterns, download_patterns, tail_file, version_number, scheduler_type, and scheduler_specs.
- Code Editor:** Displays the contents of 'dakota_remote.setup'. The code defines 'setup_properties' with the following values:

```
1 setup_properties {  
2  
3   runtime_command = "/home/raq/workbench/rte/entry.sh /home/raq/workbench/rte/dakota.py"  
4   application_path = "/projects/dakota/bin/dakota.sh"  
5   hostname = "hpc.example.gov"  
6   username = "raq"
```
- Output Console:** Shows the execution log for 'Setup_remote - Setup_remote 1'. The log indicates that the setup process completed successfully on Mon Nov 18 14:15:19 2019, with a return code of 0 and a duration of 17 seconds. The log includes steps such as uploading to the remote host, verifying transmission paths, removing existing application resources, writing the class file, reloading the module, and downloading the options file.

The status bar at the bottom shows 'Line: 41, Col: 7' and 'Validation Messages'.

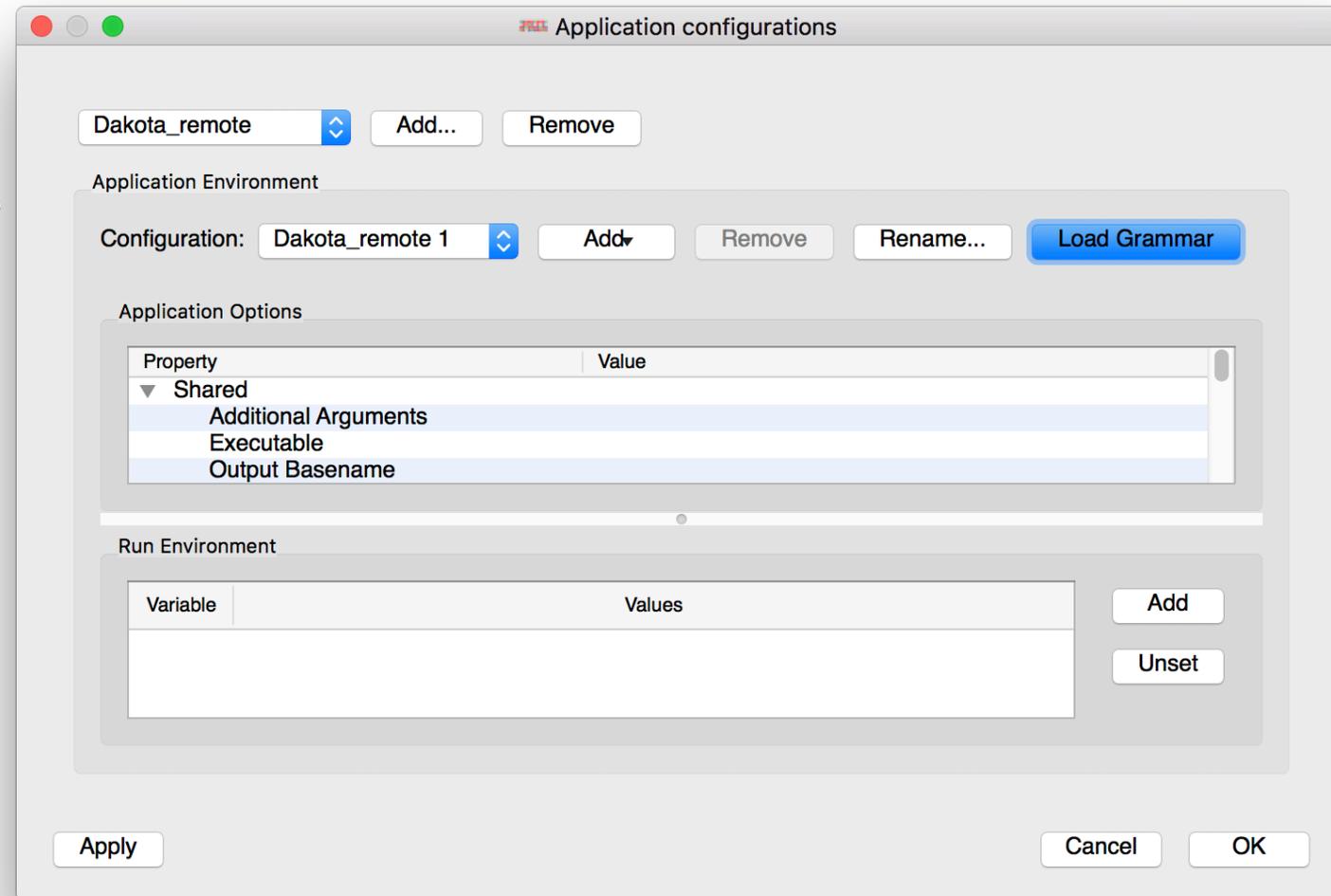
Configure the **Dakota_remote** application

- Click **File>Configurations...**
- Click **Add...** and select **Dakota_remote** and click **OK**



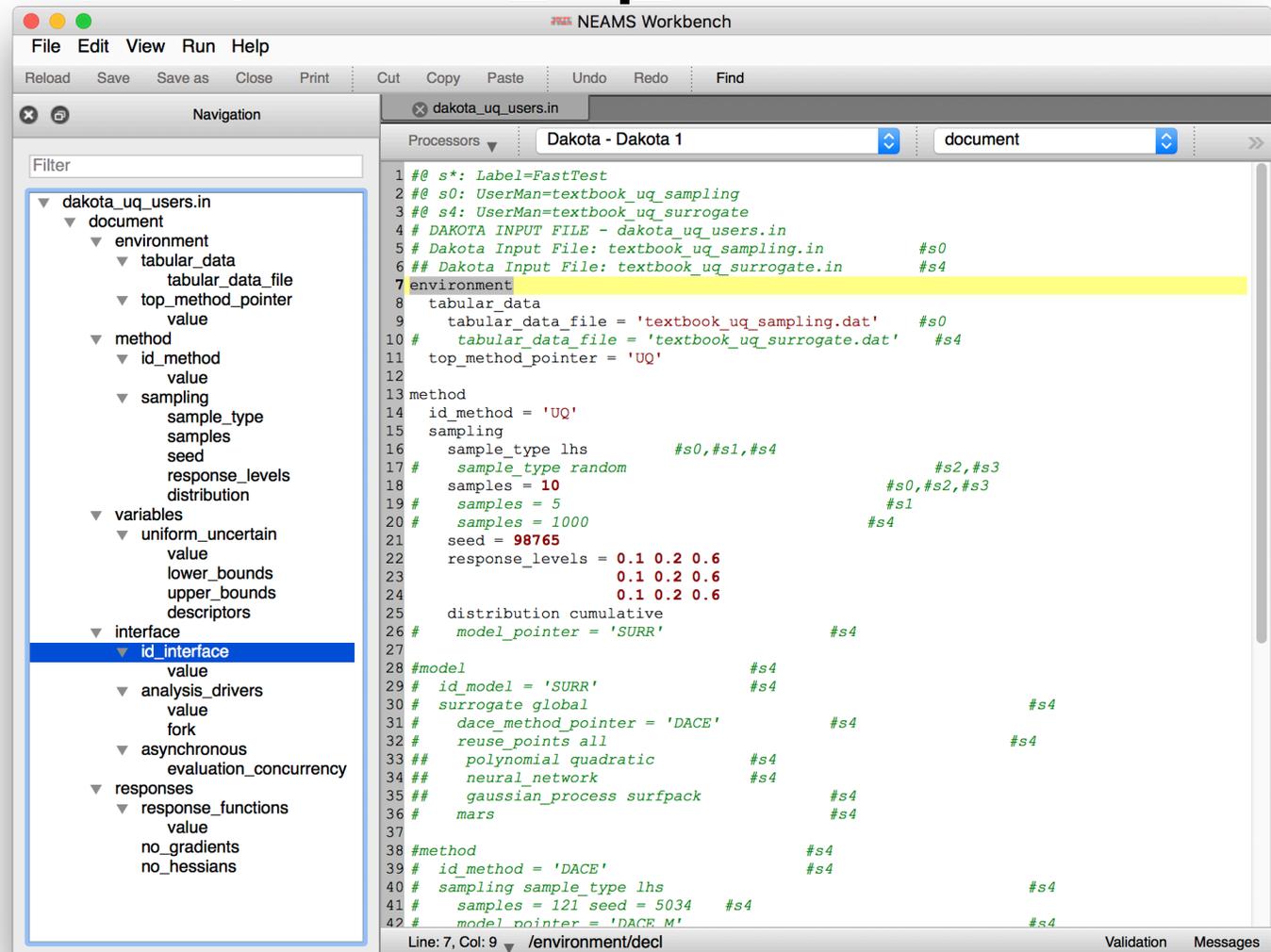
Configure the **dakota_remote** application

- Click **Load Grammar**, **Apply**, and **OK**. This enables Dakota input creation.
- **Note:** Load grammar may pause for a moment while Workbench retrieves the grammar files from the server.



Open the Dakota UQ Input

1. Click **File>Open...** and navigate to and open the **Dakota/share/dakota/test/dakota_uq_users.in**
2. select the **Dakota** local run configuration



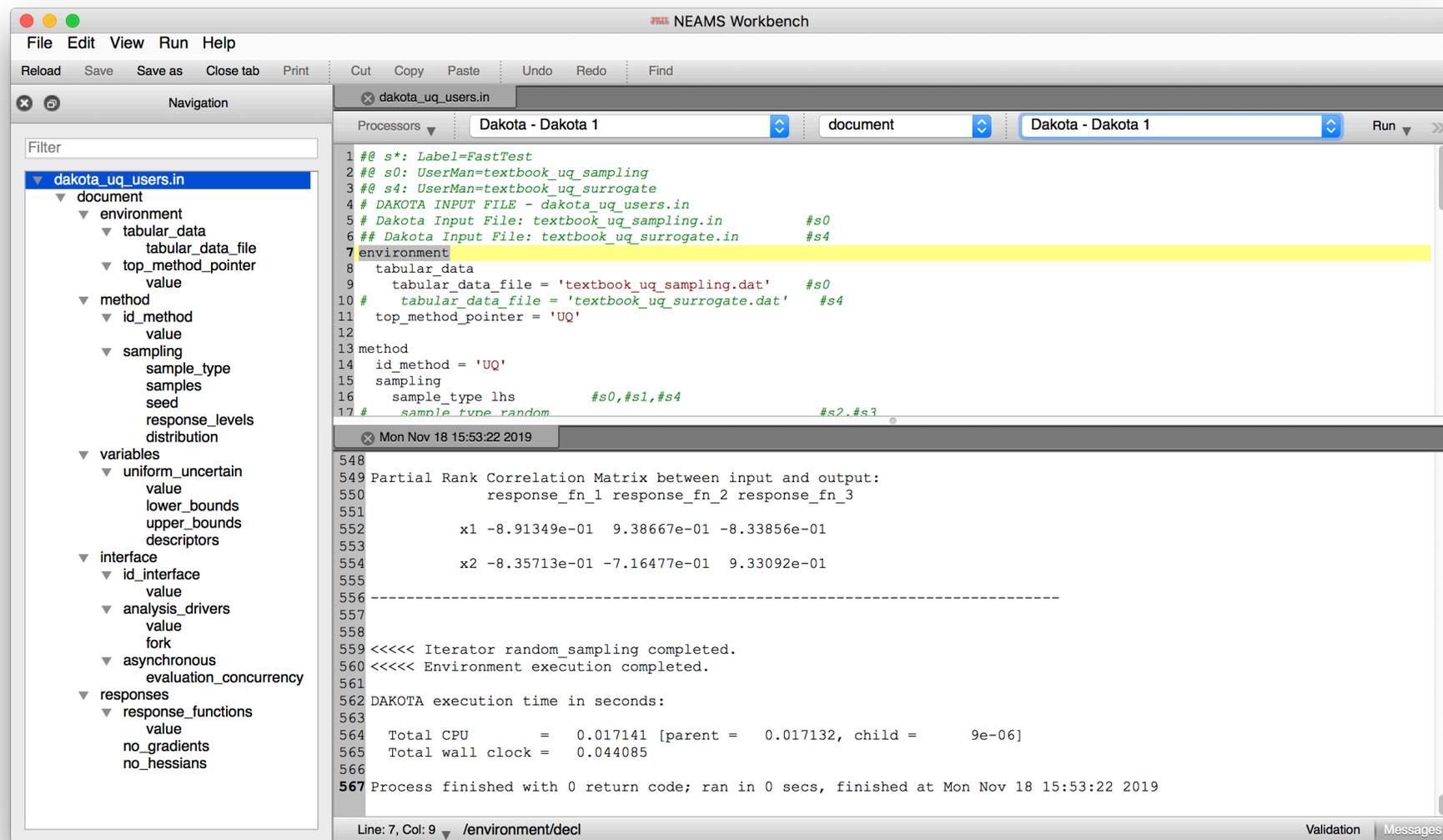
The screenshot shows the NEAMS Workbench interface. On the left, a navigation tree is expanded to show the configuration for 'dakota_uq_users.in'. The 'id_interface' is selected. The main window displays the input file 'dakota_uq_users.in' with the following content:

```
1 #@ s*: Label=FastTest
2 #@ s0: UserMan=textbook_uq_sampling
3 #@ s4: UserMan=textbook_uq_surrogate
4 # DAKOTA INPUT FILE - dakota_uq_users.in
5 # Dakota Input File: textbook_uq_sampling.in #s0
6 ## Dakota Input File: textbook_uq_surrogate.in #s4
7 environment
8 tabular_data
9 tabular_data_file = 'textbook_uq_sampling.dat' #s0
10 # tabular_data_file = 'textbook_uq_surrogate.dat' #s4
11 top_method_pointer = 'UQ'
12
13 method
14 id_method = 'UQ'
15 sampling
16 sample_type lhs #s0,#s1,#s4
17 # sample_type random #s2,#s3
18 samples = 10 #s0,#s2,#s3
19 # samples = 5 #s1
20 # samples = 1000 #s4
21 seed = 98765
22 response_levels = 0.1 0.2 0.6
23 # 0.1 0.2 0.6
24 # 0.1 0.2 0.6
25 distribution cumulative
26 # model_pointer = 'SURR' #s4
27
28 #model #s4
29 # id_model = 'SURR' #s4
30 # surrogate global #s4
31 # dace_method_pointer = 'DACE' #s4
32 # reuse_points all #s4
33 ## polynomial quadratic #s4
34 ## neural_network #s4
35 ## gaussian_process surpack #s4
36 # mars #s4
37
38 #method #s4
39 # id_method = 'DACE' #s4
40 # sampling sample_type lhs #s4
41 # samples = 121 seed = 5034 #s4
42 # model_pointer = 'DACE M' #s4
```

Line: 7, Col: 9 /environment/dec

Run the Dakota UQ Input Locally

- 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



The screenshot displays the NEAMS Workbench interface. On the left, a navigation tree shows the file structure for 'dakota_uq_users.in', including sections for environment, method, variables, interface, and responses. The main window shows the input file content, which is a Dakota input script. The script defines variables for input files and sets the method to 'UQ'. The output panel at the bottom shows the execution results, including a Partial Rank Correlation Matrix and execution statistics.

```
1 #@ s*: Label=FastTest
2 #@ s0: UserMan=textbook_uq_sampling
3 #@ s4: UserMan=textbook_uq_surrogate
4 # DAKOTA INPUT FILE - dakota_uq_users.in
5 # Dakota Input File: textbook_uq_sampling.in #s0
6 ## Dakota Input File: textbook_uq_surrogate.in #s4
7 environment
8   tabular_data
9     tabular_data_file = 'textbook_uq_sampling.dat' #s0
10    tabular_data_file = 'textbook_uq_surrogate.dat' #s4
11   top_method_pointer = 'UQ'
12
13 method
14   id_method = 'UQ'
15   sampling
16     sample_type lhs #s0,#s1,#s4
17     sample_type random #s2.#s3
```

```
548
549 Partial Rank Correlation Matrix between input and output:
550     response_fn_1 response_fn_2 response_fn_3
551
552     x1 -8.91349e-01  9.38667e-01 -8.33856e-01
553
554     x2 -8.35713e-01 -7.16477e-01  9.33092e-01
555
556 -----
557
558
559 <<<<< Iterator random_sampling completed.
560 <<<<< Environment execution completed.
561
562 DAKOTA execution time in seconds:
563
564 Total CPU      = 0.017141 [parent = 0.017132, child = 9e-06]
565 Total wall clock = 0.044085
566
567 Process finished with 0 return code; ran in 0 secs, finished at Mon Nov 18 15:53:22 2019
```

Run the Dakota UQ Input Remotely

- Select the **Dakota_remote** run configuration
- 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 message panel states **Successfully downloaded dakota_uq_users.out**

Open associated output

- Right click the **dakota_uq_users.in** file and select **Open associated files** > **dakota_uq_users.out** and review output

The image shows two screenshots of the NEAMS Workbench interface. The left screenshot shows the file explorer for 'dakota_uq_users.in' with a right-click context menu open. The 'Open associated files' option is selected, and a sub-menu is visible showing 'dakota_uq_users.out' as the chosen file. The right screenshot shows the 'dakota_uq_users.out' file open in the editor, displaying the output of the Dakota simulation, including version information, start time, and a Partial Rank Correlation Matrix.

```
10 # tabular_data_file = 'textbook_uq_surrogate.dat' #s4
11 top_method_pointer = 'UQ'
12
13 lhs #s0,#s1,#s4
14
15 response_level #s0,#s1,#s4
16
17 distribution cumulative
18
19 # model_pointer = 'SURR' #s4
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
```

```
548
549 Partial Rank Correlation Matrix between input and output:
550 response_fn_1 response_fn_2 response_fn_3
551
552 x1 -8.91349e-01 9.38667e-01 -8.33856e-01
553
554 x2 -8.35713e-01 -7.16477e-01 9.33092e-01
555
556
557
558
559 <<<<< Iterator random_sampling completed.
560 <<<<< Environment execution completed.
561
562 DAKOTA execution time in seconds:
563
564 Total CPU = 0.017141 [parent = 0.017132, child =
565 Total wall clock = 0.044085
566
567 Process finished with 0 return code; ran in 0 secs, finished at Mon Nov 18 15:53:22 2019
```

Plot Correlation Matrices

- In the *.out editor view, select the correlation matrix to plot via the **Processors>Partial correlation matrix between input and output**

The screenshot displays the NEAMS Workbench interface. The 'Processors' menu is open, and the option 'Partial correlation matrix between input and output' is highlighted with a red circle. Below the menu, the code editor shows the output of the processor, including a 'Simple Rank Correlation Matrix among all inputs and outputs' and a 'Partial Rank Correlation Matrix between input and output'. To the right, three plots are shown, each with a 'Plot' button and a 'Table' button. The top plot is the 'Simple correlation matrix (all inputs and outputs)', the middle plot is the 'Partial correlation matrix between input and output', and the bottom plot is the 'Partial rank correlation matrix between input and output'. Each plot is a heatmap showing the correlation between variables.

Summary

- You have successfully configured and run Dakota through the NEAMS Workbench via a local and remote configuration

Questions?

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