# Workshop – Installing Sandia Dakota on Windows and Red Hat Linux 7

AN MSC NASTRAN UNCERTAINTY QUANTIFICATION TUTORIAL



# Goal: Install Sandia Dakota on Windows and Red Hat Linux 7

#### Windows

#### Red Hat Linux 7

C:\Windows\system32\cmd.exe	nectarine@vm-rhel-1:~/Downloads _ 🗆 🗙
C:\Users\caparici\dakotaversion	File Edit View Search Terminal Help
Repository revision 262961b80 (2023-11-10) built Nov 10 2023 09:35:52.	[nectarine@vm-rhel-1 Downloads]\$ dakotaversion
C:\Users\caparici>	Repository revision 262961b80 (2023-11-10) built Nov 10 2023 09:23:27. [nectarine@vm-rhel-1 Downloads]\$



### Contact me

- Nastran SOL 200 training
- Nastran SOL 200 questions
- Structural or mechanical optimization questions
- Access to the SOL 200 Web App

christian@ the-engineering-lab.com



## Tutorial



## SOL 200 Web App Capabilities

#### Compatibility

- Google Chrome, Mozilla Firefox or Microsoft Edge Installable on a company laptop, workstation or
- Windows and Red Hat Linux

server. All data remains within your company.

The Post-processor Web App and HDF5 Explorer are free to MSC Nastran users.

#### Benefits

entries.

- REAL TIME error detection. 200+
- error validations.
- REALT TIME creation of bulk data
- Web browser accessible
- Free Post-processor web apps
- +80 tutorials

#### Web Apps



Web Apps for MSC Nastran SOL 200 Pre/post for MSC Nastran SOL 200. Support for size, topology, topometry, topography, multi-model optimization.



Shape Optimization Web App Use a web application to configure and perform shape optimization.



Machine Learning Web App Bayesian Optimization for nonlinear response optimization (SOL 400)



**Remote Execution Web App** Run MSC Nastran jobs on remote Linux or Windows systems available on the local network



**PBMSECT Web App** Generate PBMSECT and PBRSECT entries graphically



**Dynamic Loads Web App** Generate RLOAD1, RLOAD2 and **DLOAD** entries graphically



Ply Shape Optimization Web App Optimize composite ply drop-off locations, and generate new **PCOMPG** entries



Stacking Sequence Web App Optimize the stacking sequence of composite laminate plies



browser on Windows and Linux



HDF5 Explorer Web App Create graphs (XY plots) using data from the H5 file



## Installing Sandia Dakota on Windows



## Download Dakota on Windows

- Click the indicated link to download Dakota
- Extract the contents of the ZIP file to: C:\Program Files\Dakota

 By default, the Windows operating system limits paths to 256 characters. The original file names are very long and may exceed the 256 character limit, so use a shorter path when possible, e.g. C:\Program Files\Dakota.





## Test Dakota

1. Run this in the command prompt:

"C:\Program Files\Dakota\bin\dakota.exe"

2. Dakota is successfully configured if the output appears as shown, i.e. information regarding usage and options are displayed.

This is the end of installing Dakota on Windows.

🖾 C:\Windows\system32\cmd.exe
C:\Users\caparici>"C:\Program Files\Dakota\bin\dakota.exe" usage: C:\Program Files\Dakota\bin\dakota.exe [options and <args>] -help (Print this summary) -version (Print DAKOTA version number) -input &lt;\$val&gt; (REQUIRED DAKOTA input file \$val) -preproc [\$val] (Pre-process input file with pyprepro or tool \$val) -output &lt;\$val&gt; (Redirect DAKOTA standard output to file \$val) -parser &lt;\$val&gt; (Redirect DAKOTA input file) -no_input_echo (Do not echo DAKOTA input file) -check (Perform input checks) -put [\$val] (Perform pre-run (variables generation) phase) -run [\$val] (Perform post-run (final results) phase) -read_restart [\$val] (Read an existing DAKOTA restart file \$val) -stop_restart {\$val} (\$top restart file processing at evaluation \$val)</args>
Missing input file command line argument.
C:\Users\caparici>



The SOL 200 Web App expects Dakota to be accessible in one of the following ways:

- "C:\Program Files\Dakota\bin\dakota.exe"
  - Dakota was placed in the default location C:\Program Files\Dakota\bin\

dakota

- The location of the dakota.exe file was added to the PATH environment variable.
- The location of dakota.exe may be any non-default location

#### > "C:\Program Files\Dakota\bin\dakota.exe"

- - X C:\Windows\system32\cmd.exe C:\Users\caparici>"C:\Program Files\Dakota\bin\dakota.exe" usage: C:\Program Files\Dakota\bin\dakota.exe [options and <args>] C:\Program Files\Dakota\bin\dakota.exe [options and <args>] -help (Print this summary) -version (Print DAKOTA version number) -input <\$val> (ReGUIRED DAKOTA input file \$val) -preproc [\$val] (Pre-process input file with pyprepro or tool \$val) -output <\$val> (Redirect DAKOTA standard output to file \$val) -output <\$val> (Redirect DAKOTA standard error to file \$val) -preser <\$val> (Redirect DAKOTA standard error to file \$val) -no\_input\_echo (Do not echo DAKOTA input file) -check (Perform input checks) -pre\_run [\$val] (Perform pre-run (variables generation) phase) -run [\$val] (Perform post-run (final results) phase) -read\_restart [\$val] (Read an existing DAKOTA restart file \$val) -stop\_restart [\$val] (Write a new DAKOTA restart file \$val)

Missing input file command line argument.

C:\Users\caparici>

#### > dakota





This exercise configured Dakota v6.19.0.

Other versions of Dakota may be installed.

- Click the indicated link
- Navigate to the version of interest
- Click Assets
- Download one of the files whose name contains the following strings
  - .zip
  - Windows
  - cli or gui cli

One ZIP file is needed. The string cli indicates the ZIP file contains the Dakota solver. The string *qui* indicates the ZIP file contains the GUI.

https://github.com/snl-dakota/dakota/releases

(1)



(2)





 Note that different options (cli or gui) are available for each version. For example, version 6.20.0 only provides the solver (cli) for Dakota on Windows. There is no GUI (gui) option for Windows.

See file:

dakota-6.20.0-publicwindows.Windows.x64-cli.zip

#### May 13 dakota-snl v6.20.0

-0- 494027b



#### Release 6.20.0

Documentation and Release Notes

File name key:

- $\tt gui\_cli$  contain both the Dakota GUI and command-line executable
- cli Command-line executable only
- gui GUI only
- src Source package, .zip for Windows line endings and .tar.gz for POSIX

Assets 14		
	35.2 MB	May 13
	643 MB	May 13
	604 MB	May 13
	421 MB	May 13
Gdakota-6.20.0-public-rhel8.Linux.x86_64-cli.tar.gz	60.4 MB	May 13
	481 MB	May 13
⊗dakota-6.20.0-public-src-cli.tar.gz	140 MB	May 13
	154 MB	May 13
	6.5 MB	May 13
	140 MB	May 13
Source code (zip)		Apr 29
Source code (tar.gz)		Apr 29

Questions? Email: christian@ the-engineering-lab.com

(1)



## Installing Sandia Dakota on Red Hat Linux 7



#### https://github.com/snl-dakota/dakota/releases/download/v6.19.0/dakota-6.19.0-public-rhel7.Linux.x86\_64-cli.tar.gz (1)

## Download Dakota on Windows

- Click the indicated link to download Dakota
- Extract the contents of the ZIP file to: /opt/dakota. Use the indicated commands.



tar -xzf dakota-6.19.0-public-rhel7.Linux.x86\_64-cli.tar.gz

sudo mv dakota-6.19.0-public-rhel7.Linux.x86\_64-cli /opt/dakota

2



## Test Dakota

1. Run this in the command prompt:

#### /opt/dakota/bin/dakota

2. Dakota is successfully configured if the output appears as shown, i.e. information regarding usage and options are displayed.

This is the end of installing Dakota on Red Hat Linux 7.

nectarine@vm-rhel-1:~/Downloads × File Edit View Search Terminal Help [nectarine@vm-rhel-1 Downloads]\$ /opt/dakota/bin/dakota(1) usage: dakota [options and <args>] -help (Print this summary) (2) -version (Print DAKOTA version number) -input <\$val> (REQUIRED DAKOTA input file \$val) -preproc [\$val] (Pre-process input file with pyprepro or tool \$val) -output <\$val> (Redirect DAKOTA standard output to file \$val) -error <\$val> (Redirect DAKOTA standard error to file \$val) -parser <\$val> (Parsing technology: nidr[strict][:dumpfile]) -no input echo (Do not echo DAKOTA input file) -check (Perform input checks) -pre run [\$val] (Perform pre-run (variables generation) phase) -run [\$val] (Perform run (model evaluation) phase) -post run [\$val] (Perform post-run (final results) phase) -read restart [\$val] (Read an existing DAKOTA restart file \$val) -stop restart <\$val> (Stop restart file processing at evaluation \$val) -write restart [\$val] (Write a new DAKOTA restart file \$val)

Missing input file command line argument.



The SOL 200 Web App expects Dakota to be accessible in one of the following ways:

- /opt/dakota/bin/dakota
  - Dakota was placed in the default location /opt/dakota
- dakota
  - The location of the dakota executable file was added to the PATH environment variable.
  - The location of dakota may be any non-default location

#### \$ /opt/dakota/bin/dakota

nectarine@vm-rhel-1:~/Downloads _ 🗖	×
File Edit View Search Terminal Help	
<pre>Inter Edit View Scatch Verminit Prep [nectarine@vm-rhel-1 Downloads]\$ /opt/dakota/bin/dakota usage: dakota [options and <args>] -help (Print this summary) -version (Print DAKOTA version number) -input &lt;\$val&gt; (REQUIRED DAKOTA input file \$val) -preproc [\$val] (Pre-process input file with pyprepro or tool \$val) -output &lt;\$val&gt; (Redirect DAKOTA standard output to file \$val) -output &lt;\$val&gt; (Redirect DAKOTA standard error to file \$val) -error &lt;\$val&gt; (Redirect DAKOTA standard error to file \$val) -parser &lt;\$val&gt; (Parsing technology: nidr[strict][:dumpfile]) -no_input_echo (Do not echo DAKOTA input file) -check (Perform input checks) -pre_run [\$val] (Perform pre-run (variables generation) phase) -run [\$val] (Perform post-run (final results) phase)</args></pre>	
<pre>-read_restart [\$val] (Read an existing DAKOTA restart file \$val) -stop_restart &lt;\$val&gt; (Stop restart file processing at evaluation \$val) vrite restart [\$val] (Write a pay DAKOTA restart file \$val)</pre>	

Missing input file command line argument.

#### \$ dakota

•				I	nectarin	e@vm-rh	hel-1:	~/Do\	wnload	ds				-		×
File	Edit	View	Search	Terminal	Help											
[nec usag	tarir e: da - - - - - - - - - - - - - - - - - - -	e@vm- kota help versi input prepr outpu error parse no_ir check pre_r run [ post_ read_ stop_ write	<pre>rhel-1 [optio (Print on (Pr &lt;&lt;\$val oc [\$v t &lt;\$val &lt;&lt;\$val out &lt;\$val out ect (Perf un [\$v \$val] run [\$v restar restar </pre>	Downloa ns and < this su int DAKO > (REQUI al] (Pre l> (Redi > (Redi > (Redi ) (Per d) (Per sho (Do n orm inpu al] (Per (Perform val] (Per t [\$val] t <\$val>	ds]\$ d args>] mmary) TA ver RED DA -proce rect DA ect DA ing te ot ech t chec form p run ( rform (Read (Stop ] (Wri	sion no KOTA in ss inpo AKOTA s chnolog o DAKO ks) re-run model o post-ru an ex: restan te a no	umbe nput stan gy: TA i (va eval un ( isti	er) ile idard ard nidr nput uati fina ile AKOT	e \$va with outp erron [str: file les g on) p l res AKOTA proce	al) pyp put r to ict] e) gene bhas sult A re essi star	repro to fi file [:dum ratio e) s) ph start ng at t fil	o or le \$ sva pfil n) p ase) i fil c eva e \$v	tool sval) al) .e]) bhase) .e \$va aluati val)	\$val ) al) ion \$	.) Sval	)

Missing input file command line argument.



The PATH environment variable may be modified as follows.

1. Run

sudo gedit ~/.bashrc

2. Append :/opt/dakota/bin to the PATH variable as shown.

Example 1

Before: export PATH=\$PATH

After: export PATH=\$PATH:/opt/dakota/bin

Example 2

Before: export PATH=\$PATH:/msc/MSC\_Nastran/20181/bi n

After: export PATH=\$PATH:/msc/MSC\_Nastran/20181/bi n:/opt/dakota/bin

3. Save the file

- 4. Close and reopen the terminal
- 5. Dakota may now be accessed with the shorthand form:

ak	ota

Open 👻	Æ	.bashrc /home/nectarine	Save	≡	-		×
# .bashrc							
# Source gl if [ -f /et . / fi	l <mark>oba</mark> tc/ba /etc,	l definitions ashrc ]; then /bashrc					
# Uncomment # export Si	t th (STE	e following line if you don't like systemctl's auto MD_PAGER=	-paging	feat	ure:		
export MSC_		ENSE_FILE="27500@192.168.56.1"					
<pre># You will # This is r # "The resu sessions. export PATH</pre>	see norma ult : Thi: H=\$PA	duplicates when you do echo \$PATH al, per <u>https://bugzilla.redhat.com/show_bug.cgi?id</u> is two duplicate user PATH entries in \$PATH, gets e s does no harm, but may confuse users." ATH:/msc/MSC_Nastran/20181/bin:/opt/dakota/bin	<u>=1652639</u> ven more	2 e wit	h nest	ed b	ash



This exercise configured Dakota v6.19.0.

Other versions of Dakota for Red Hat Linux 7 or 8 may be installed.

- 1. Click the indicated link
- 2. Navigate to the version of interest
- B. Click Assets
- 4. Download one of the files whose name contains the following strings
  - .tar.gz
  - rhel7 or rhel8
  - cli or gui\_cli

One tar.gz file is needed. The string *cli* indicates the ZIP file contains the Dakota solver. The string *gui* indicates the ZIP file contains the GUI. The string *rhel7* indicates compatibility with Red Hat Linux 7. The string *rhel8* indicates compatibility with Red Hat Linux 8.

#### https://github.com/snl-dakota/dakota/releases

(1)





1. Note that different options (CLI or GUI, RHEL 7 or 8) are available for each version. For example, version 6.20.0 only provides Dakota for Red Hat Linux 8, not 7. See file:

dakota-6.20.0-public-rhel8.Linux.x86\_64cli.tar.gz

dakota-6.20.0-public-rhel8.Linux.x86\_64gui\_cli.tar.gz

If your operating system is not listed, you may compile your own version of Dakota using the source files that are available, e.g. dakota-6.20.0-public-src-cli.tar.gz.

Contact The Engineering Lab for assistance.

#### May 13 dakota-snl V6.20.0 --- 494027b

Compare +



#### Release 6.20.0

Documentation and Release Notes

File name key:

- $\tt gui\_cli$  contain both the Dakota GUI and command-line executable
- cli Command-line executable only
- gui GUI only
- src Source package, .zip for Windows line endings and .tar.gz for POSIX

	35.2 MB	May 13
	643 MB	May 1
∲dakota-6.20.0-public-Darwin.x86_64-gui.tar.gz	604 MB	May 1
∲dakota-6.20.0-public-Linux.x86_64-gui.tar.gz	421 MB	May 1
	60.4 MB	May 1
Gdakota-6.20.0-public-rhel8.Linux.x86_64-gui_cli.tar.gz	481 MB	May 1
∲dakota-6.20.0-public-src-cli.tar.gz	140 MB	May 1
∲dakota-6.20.0-public-src-cli.zip	154 MB	May 1
∲dakota-6.20.0-public-src-gui.zip	6.5 MB	May 1
	140 MB	May 1
Source code (zip)		Apr 2
Source code (tar.gz)		Apr 2

Questions? Email: christian@ the-engineering-lab.com

1



End of Tutorial

