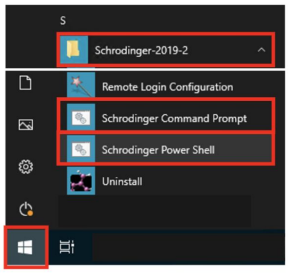
Using the Command Line with the Schrödinger Platform

For more information, visit our Knowledge Base article on ***schrodinger.com/kb/1842***

Uses most (but not all) Unix commands, and backslash with paths

Setup Schrödinger Variable

1. From the **Start** button, navigate to your **Schrodinger** installation directory and expand the directory list
2. Click on **Schrodinger Command Prompt**, or **Schrodinger Power Shell**
   - A console will open on your default Working Directory in **HOME > Documents > Schrodinger** location
   - This console will have all Schrodinger's environment variables defined



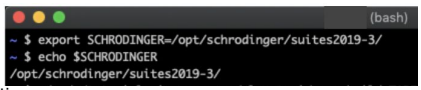
Windows Prompt File Commands

cd	Changes current location to a specified directory.
ls	Lists the files and subdirectories.
pwd	Shows the current working directory.
echo	Displays the text after the echo.
rm	Deletes file or directories.
mv	Moves an item from one location to another. Also used for renaming files or directories.
cp	Copies an item from one location to another.
cat	Displays the entire content of a specified file. Use it with **-Head 5** to display first 5 lines. Use it with **-Tail 5** to display last 5 lines. Replace 5 with the number of lines you want to display.
set	To create environment variables using `set <env name>=<variable>`
sh	To open a Unix virtual environment.

Uses all Unix commands, and forward slash with paths

Setup Schrödinger Variable

1. Open your favorite terminal application
2. Prepare your Schrödinger environment variable by pointing to the Schrödinger suite installation path,
   e.g.: `export SCHRODINGER=/opt/schrodinger/suites*/`
   where suites* is the version of the distribution installed
   - The example above is the default installation location on Mac platform.
3. To use an environment variable, add **dollar sign** before the variable name `$SCHRODINGER`



Unix File Commands

cd	Changes current location to a specified directory.
ls	Lists the files and subdirectories.
pwd	Shows the current working directory.
echo	Displays the text after the echo. Try it with **$SCHRODINGER**.
rm	Deletes file. Use it with **-r** to delete directories and subdirectories.
mv	Moves an item from one location to another. Also used for renaming files or directories.
cp	Copies a file from one location to another. Use it with **-r** to copy directories and subdirectories.
cat	Displays the entire content of a specified file.
head, tail	Display first or last 10 lines for a file respectively.
export (for csh: setenv)	To create environment variables using `export <env name>=<variable>`

Tips for all platforms

SSH (Unix, Windows 10)

Secure Shell gives a secure way to access a remote computer.

1. Get remote computer IP address
2. Get a remote username (often is your own)
3. Launch terminal
4. Type `ssh username@ip_address`
5. Authenticate your credentials.

Sending Files Remotely

- **rsync (Unix only):**
  - Type `rsync -av <dir_name> username@ip_address:<destination_directory>`
    -a; copying recursively + preserves files modifications times, group, owner, and permissions.
    -v: with verbosity
- **scp:**
  - For files, type: `scp <file1_name> <file2_name> … username@ip_address:<destination_directory>`
  - For directories, type: `scp -r <dir_name> … username@ip_address:<destination_directory>`
- **WinSCP (Windows only):**
  - An open source software that provides secure file transfer from a local to a remote computer using a graphical interface..

Connect to a remote computer **Using third party software**

- From Windows to Linux/Mac:
  - PuTTY
- From Linux/Mac to Windows:
  - Microsoft Remote Desktop
- From Linux/Mac to Mac/Linux:
  - x2go, NoMachine

man (Unix only)

- To retrieve manual information of a certain command.
  e.g.: To view the ls command manual and learn more about its usage, type `man ls`

whoami

- Type `whoami` to show the username of the current user.

-h, --help

- Displays command, or **script,** usage information if exists
  e.g.: To view the rsync command usage, type `rsync -h` or `rsync --help`

<Tab>

- For autocompletion, hit Tab on your keyboard while typing a command, option, file name, or directory and the command line interface will complete what you are typing or suggest options to you.

SCHRÖDINGER®

Submitting Jobs Through the Console

Useful Schrödinger tools

jobcontrol

- The job control utility allows you to perform a number of job control tasks from the console.
- Common useful arguments:
  - `-list` to view the list of job ID that are currently running with details..
  - `-show <job_id>` to show information of the specified job ID.
  - `-kill <job_id>` to stop the specific job ID and recover output files.
  - `-view log <job_id>` to view part of the log file of the specified job ID.

utilities

- General executable applications available in the `utilities` directory of the Schrödinger installation. It contains utilities that are associated with a specific panel, and others that are useful scripts of general-purpose.

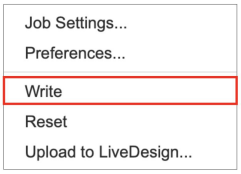
run

- To run Python scripts from the Schrödinger installation in
  - `mmshare-v* > python > common` ,or
  - `mmshare-v* > python > scripts`

  Visit **schrodinger.com/scriptcenter** to know more about scripts that come with Schrödinger suites.

Running a Maestro job from the console

1. From a Schrödinger panel, click on the **cog icon**  located at the bottom of the panel beside the Job name.
   - A drop-down menu will appear.
2. Click **Write**.
   - In most panels, a message in green font color will appear at the bottom of the panel indicating the files have been written successfully.
   - Current job files are copied to the current Working Directory.



3. Navigate through the command line:
   - Type `pwd` to check current Working Directory
   - Type `cd` to enter a job directory, or to navigate to the job files location
   - Type `ls` to list all the files in a job directory
   - Locate the file with **.sh** extension.
     - This is the job submission script that contains the shell command.
     - Type `cat` to display the file content.

View job results in Maestro:

- Import the output file to the Maestro project.

Output structure file for **Biologics** and **Small Molecule suites** commonly used modules:

- Generally *out.maegz
- Glide -> *pv.maegz
- Shape Screen -> *align.maegz
- Enumeration -> *.maegz
- Desmond -> *out.cms
- Phase -> *.phypo
- FEP -> *out.fmp

Output structure file for **Materials Science suite** commonly used modules:

- Optoelectronics -> *opto.maegz
- QM Multistage -> *out.maegz
- Bond and Ligand Dissociation -> *bde.maegz
- Thermophysical Properties -> *out.maegz
- Modules with MD trajectories -> *out.cms

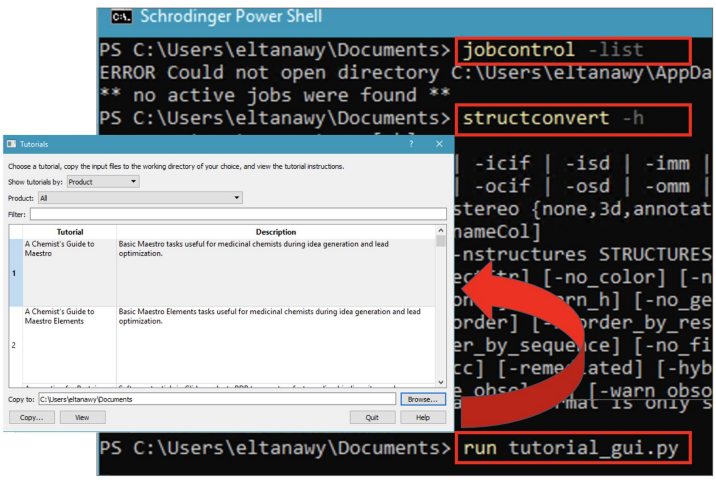
To run the job locally:

1. Do either of the following:
   A. **Copy** the content of the .sh file and **paste** it in the console.
   B. Run the .sh script itself by typing
      `run <job_name>.sh`
2. Press **Enter** to execute the command.

To run the job remotely:

A. Include `-HOST <hostname>` in the shell command in the console, or
B. 1. Use `rsync` or `scp` command to send the job files to a remote host.
   2. `ssh` to that remote host.
   3. Run the job locally as indicated in the previous section.
      - Use `<script_name> -h` to view script usage.

An example of a Windows console using Schrödinger tools:



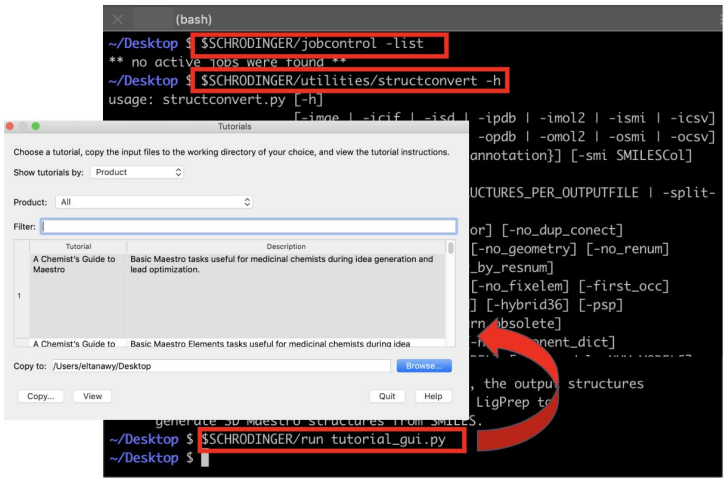
To run the job locally:

1. Do either of the following:
   A. **Copy** the content of the .sh file and **paste** it in the terminal.
   B. Run the .sh script itself by typing
      `$SCHRODINGER/run <job_name>.sh`
2. Press **Enter** to execute the command.

To run the job remotely:

A. Include `-HOST <hostname>` in the shell command in the terminal, or
B. 1. Use `rsync` or `scp` command to send the job files to a remote host.
   2. `ssh` to that remote host.
   3. Run the job locally as indicated in the previous section.
      - Use `$SCHRODINGER/<script_name> -h` to view script usage.

An example of a Unix terminal using Schrödinger tools:



SCHRÖDINGER®