

# Advanced VSim Visualization Using VisIt

Thomas G. Jenkins  
Tech-X Corporation  
September 24, 2021

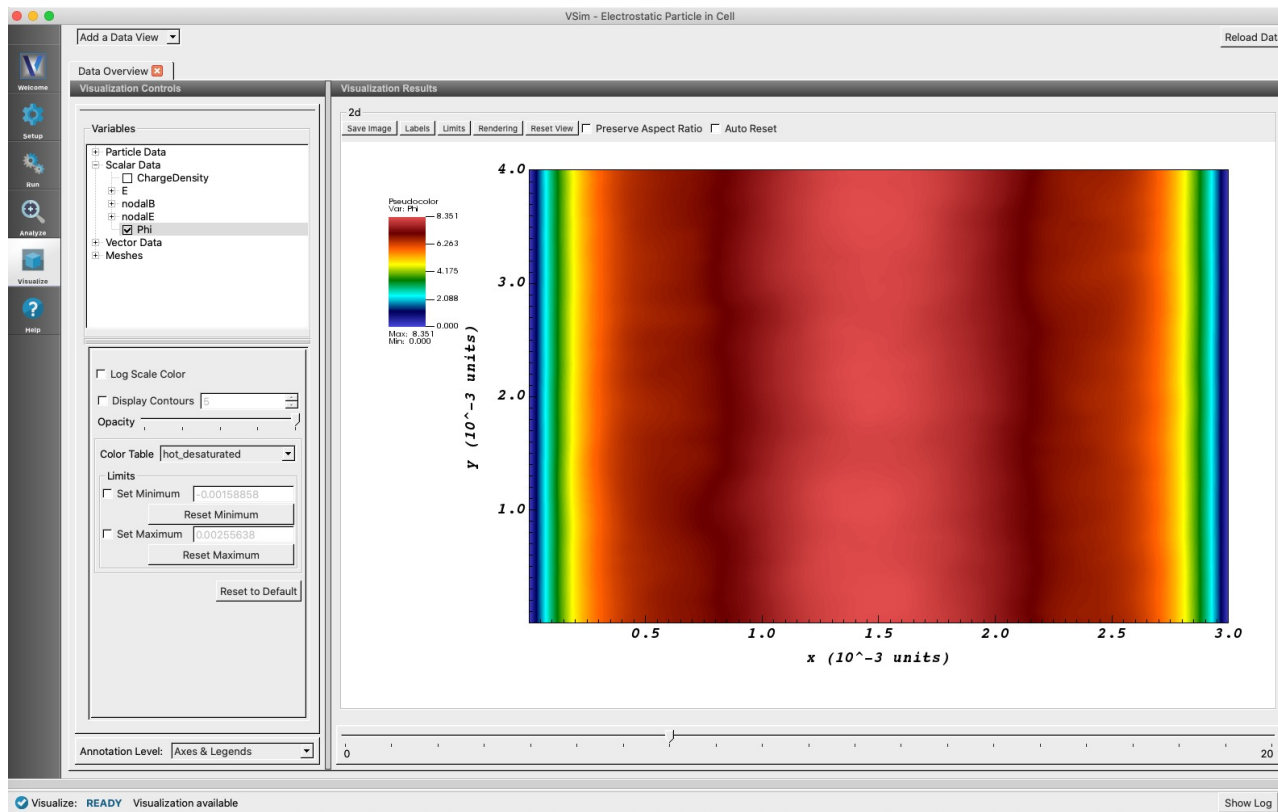
# VisIt can be used to generate high-quality plots or animations of your simulation data

On this slide, I showed the movie available at

[http://nucleus.txcorp.com/~tgjenkins/movies/ICRF\\_in\\_CMod.mov](http://nucleus.txcorp.com/~tgjenkins/movies/ICRF_in_CMod.mov)

# What is VisIt? You've already seen parts of it in VSim.

- VisIt is a (free) visualization software tool, with ongoing development supported by the U.S. Department of Energy, available at <https://visit.llnl.gov>.



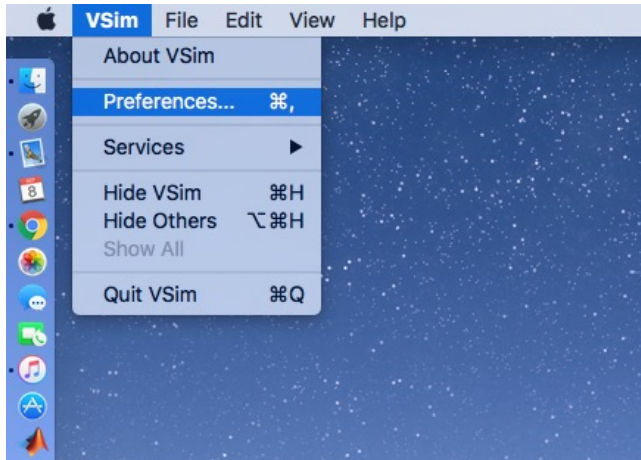
-VSim Composer includes a VisIt installation; VSim output files are formatted to be readable by VisIt.

-VSim Composer makes many common VisIt operations easily accessible to VSim users.

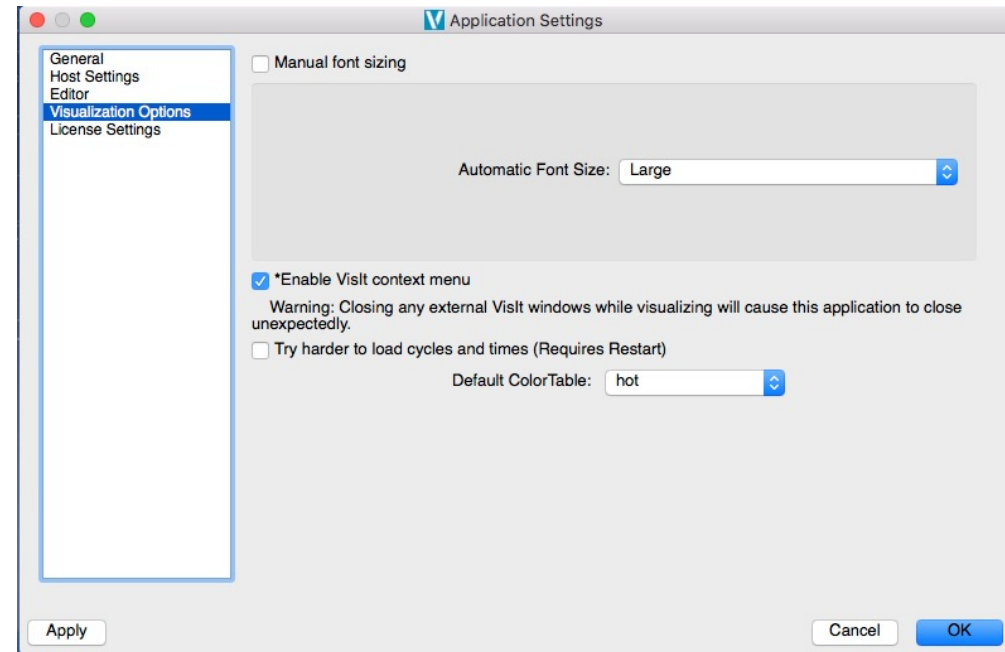
-Other VisIt features can also be accessed – will demonstrate this.

# Accessing additional VisIt features from VSim Composer: setup

- On a Mac, open VSim **Preferences** menu; on Windows, open VSim **Tools > Settings** menu:



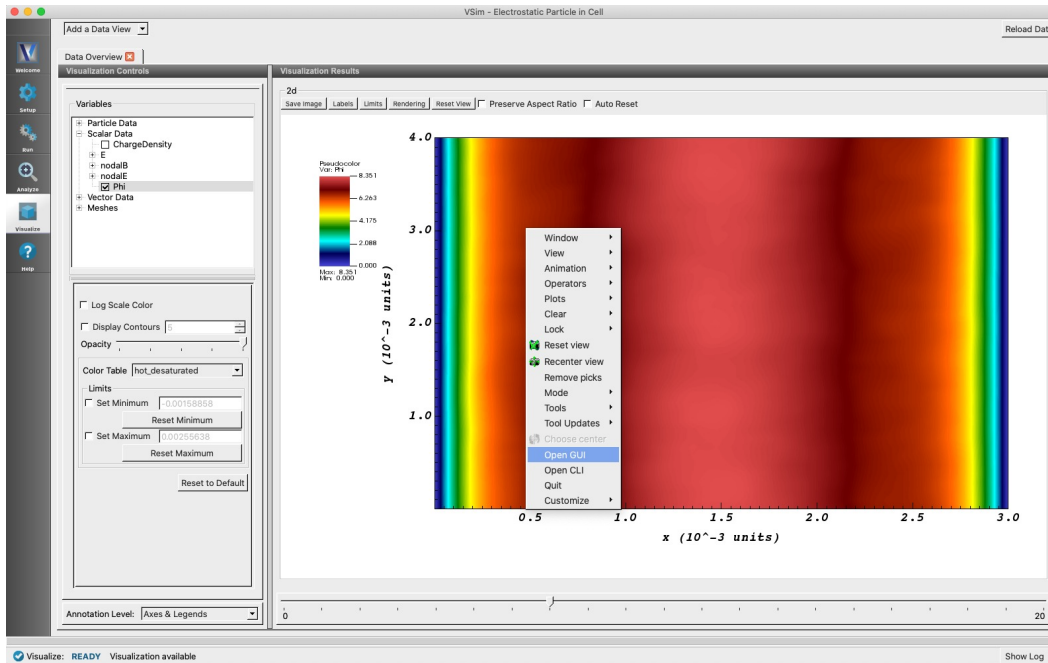
- Under the **Visualization Options** menu, check the **Enable VisIt context menu** box; then click **Apply** followed by **OK**.



- A warning will appear: closing VisIt windows will also close Composer (I'll show what that means, in practice, in a moment).

# Accessing additional VisIt features from VSim Composer: execution

-From VSim Composer's **Visualize** tab, open a plot, right-click on it, and then select the **Open GUI** option – this will launch the VisIt user interface.



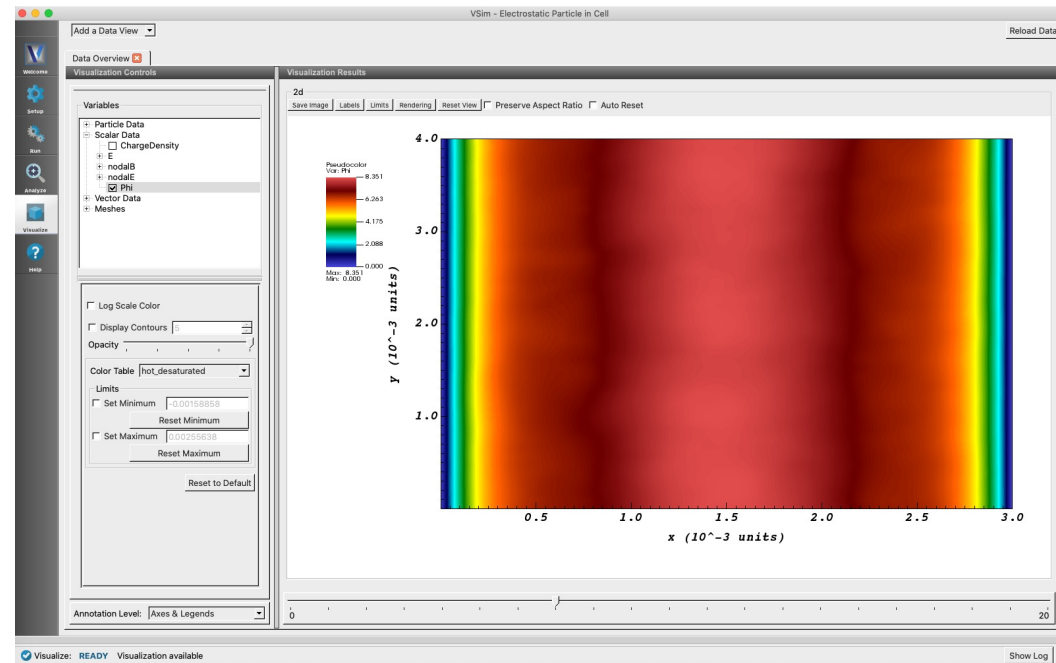
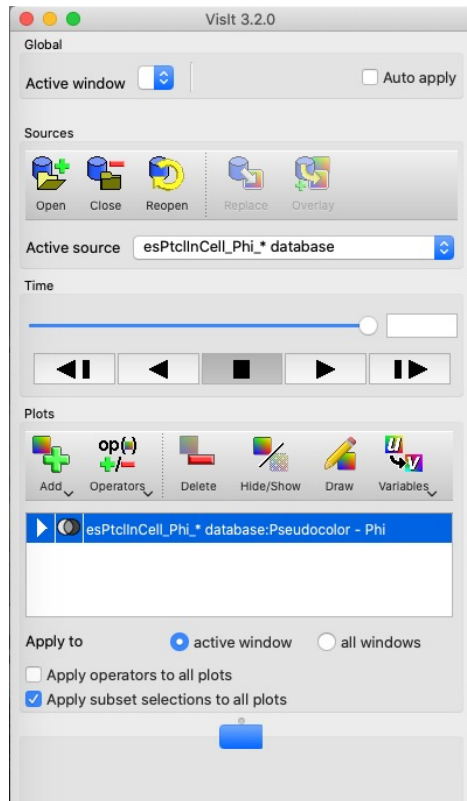
-Please report any bugs in this process – on some of the newer Mac OS's we've run into minor challenges.

-Can also download binaries for standalone VisIt installations at <https://visit.llnl.gov>.



# Now we have both a VisIt window and a Composer window

- Closing one window will close the other (recall the earlier warning)
- Plot can be modified by either window – helpful if only minor changes needed
- Composer opens all relevant datasets in separate VisIt windows



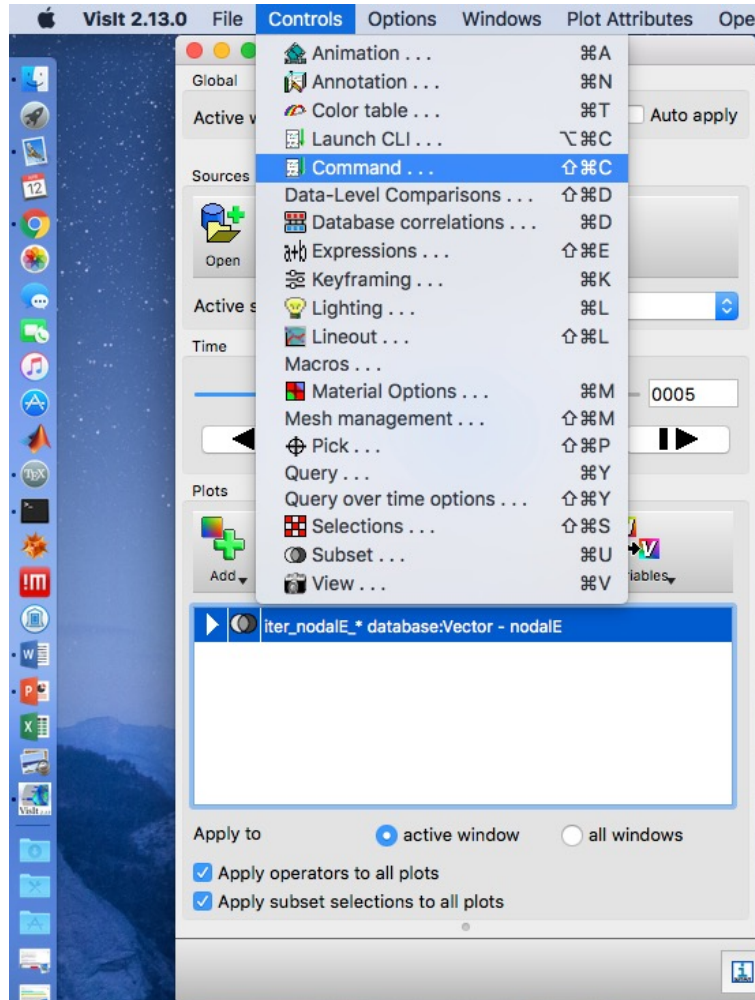
(At this point I'm going to switch to standalone VisIt, for ease in demonstration.)

# Demonstration of various VisIt operations

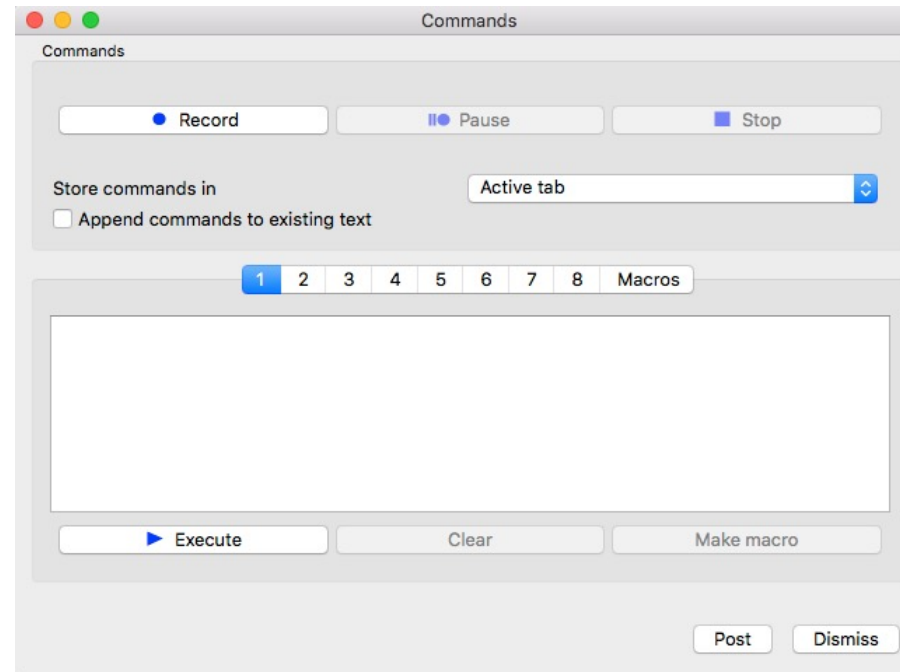
- Opening files
- Making various kinds of plots:
  - \*Contour
  - \*Volume
  - \*Pseudocolor
  - \*Vector
  - \*Scatter
  - \*History
- Operators
- Annotations
- Saving plot windows
- Saving your VisIt session
- Building expressions

In this part of the talk I did a live demo of the VisIt software

# Recording commands for later use

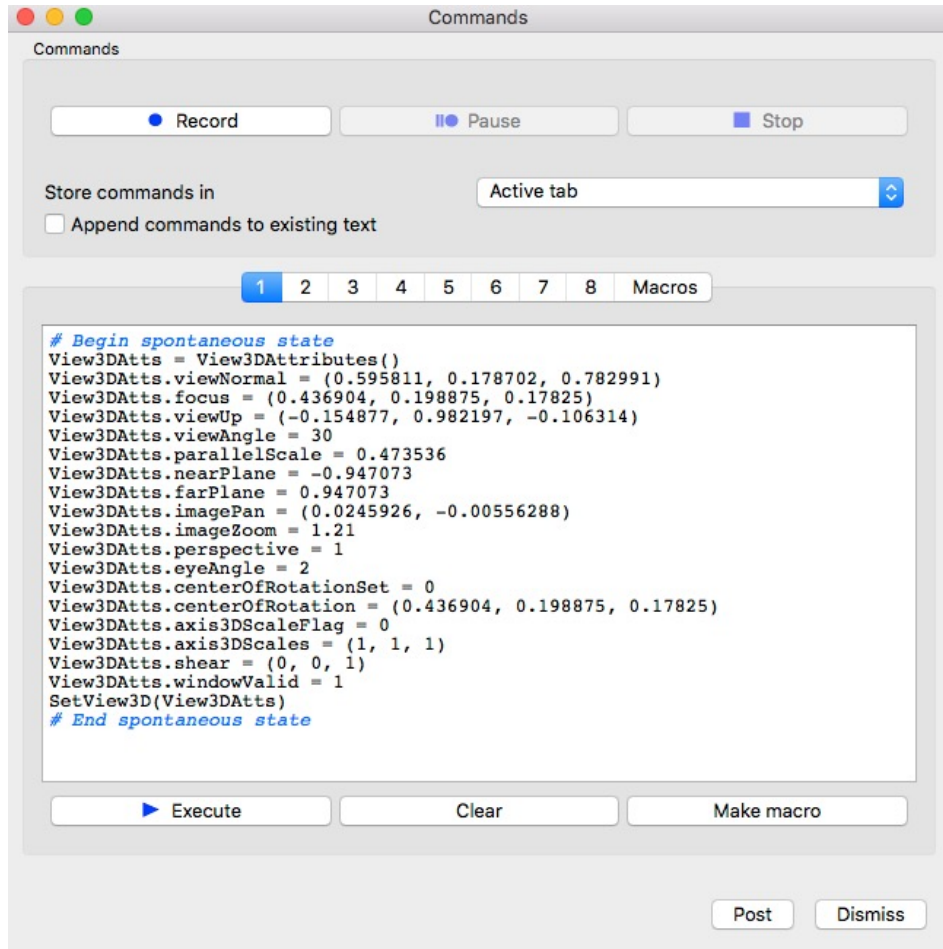


- Sometimes we want to do the same analysis on many sets of VSim data (e.g. to look at wave amplitudes in multiple runs that have different density profiles in the plasma edge).
- We can use VisIt's Command window to set up these kinds of operations.





# General principles for recording commands



**Record**, do all the things you need to do to make your plot look nice, then **Stop**.

VisIt keeps track of all the plot attributes that changed, and how the new attributes were applied to the plot. (Here, I've changed the view angle on an existing 3D plot.)

**Execute** will repeat every command in the window (useful if you want to make only minor changes). **Make Macro** will save the sequence of commands as an object you can call on later (even if you quit and restart VisIt).

# Python scripting with VisIt commands

- Recorded commands can also be copied into a python script, which can be invoked from the command line via  
[/Applications/VisIt3.2.app/Contents/Resources/bin/visit -s myScript.py](#)  
on Mac (I expect it works roughly the same way for Windows/Linux machines).
- Presumably you'll want your script to loop over timesteps, and save each timestep as a separate image. If you're doing this, you can even run the script without rendering the images on your screen:  
[/Applications/VisIt3.2.app/Contents/Resources/bin/visit -nowin -s myScript.py](#)  
(make sure you really know what your script is doing first).
- I have more details on this process in a talk from the 2018 TWSS that can be found on my website,  
[http://nucleus.txcorp.com/~tgjenkins/pres/Jenkins\\_TWSS2018\\_VisIt.pdf](http://nucleus.txcorp.com/~tgjenkins/pres/Jenkins_TWSS2018_VisIt.pdf)

# Using VisIt on remote machines

- Often we're running physics simulations on remote hosts:
  - local clusters
  - NERSC or OLCF supercomputers (edison, cori, titan)
- VisIt can be used to access data on these and other remote hosts, if we configure it properly.
  - Visualization options are chosen from your machine (laptop or desktop)
  - Computation needed to render the requested image is performed by remote host
  - Image is rendered (or saved directly to a file without rendering) on your machine
- Configuring VisIt to properly connect with other machines is not hard in principle. NERSC provides config files for its own machines, etc. Main issue: **need same version of VisIt on both machines.**
- [visitusers.org](http://visitusers.org) has useful documentation to help you get things set up.

# Some concluding thoughts

- VisIt can be a very powerful tool for visualizing your data. If you can imagine what a plot might look like, there's probably a way to make it in VisIt.
- Good plots and movies can effectively promote your work to program managers / investors / bosses / grant reviewers / etc.
- VSim data is formatted to enable VisIt usage by default.
- Python scripting, expression construction, and macro construction can be used to construct complex, repeatable data analysis workflows.
- The learning curve for VisIt can be steep, but the wiki site and user forum at <http://visitusers.org> are valuable resources.



# Useful links

Main VisIt page:

<https://visit.llnl.gov>

Visit users forum:

<https://visitusers.org>

My 2018 TWSS talk on visualization:

[http://nucleus.txcorp.com/~tgjenkins/pres/Jenkins\\_TWSS2018\\_VisIt.pdf](http://nucleus.txcorp.com/~tgjenkins/pres/Jenkins_TWSS2018_VisIt.pdf)

Examples of some movies I have made with VisIt:

<http://nucleus.txcorp.com/~tgjenkins/movies.html>