Development of Web Tools for LTX-beta
Nathaniel Sokolow (Community College of Baltimore County), advised by Robert Kaita (PPPL)

**LTX Web Tools**

This summer, development commenced on a suite of web tools (basically web applications) which could be used to plot MDSplus data. Hopefully, in the future the scope of the project will expand beyond just plotting, and may eventually be used on NSTX. See the right panel first for an overview of LTX plotting through MDS and Scripting.

**LTX Web Tools**

- **Model**: The source for the input page. Written as a Python class, in this application it provides the HTML input form in which the user writes the shotnumber and node they want. The input template renders the model as HTML for your browser to show when you first go to the site.
- **Controller**: The part of the Application that actually contains the commands to show the input template (the model); and calls the compute method. It then saves the plot output from the compute method, and adds that to the output template. This is the app the server runs.
- **Compute**: Accepts the user parameters input to it by the Controller method (from the input form), and uses them to output a plot. For LTX, this is a script which connects to the LTX tree and plots the data from the desired node to the template.

The photos below are from a different project, but the projects are conceptually the same.

**LTX Web Tools**

**Legends**

- **Model**: The source for the input page. Written as a Python class, in this application it provides the HTML input form in which the user writes the shotnumber and node they want. The input template renders the model as HTML for your browser to show when you first go to the site.
- **Controller**: The part of the Application that actually contains the commands to show the input template (the model); and calls the compute method. It then saves the plot output from the compute method, and adds that to the output template. This is the app the server runs.
- **Compute**: Accepts the user parameters input to it by the Controller method (from the input form), and uses them to output a plot. For LTX, this is a script which connects to the LTX tree and plots the data from the desired node to the template.

The photos below are from a different project, but the projects are conceptually the same.

**LTX Beta Data Acquisition**

**Overview**

- **LTX** takes several dozen plasma shots per day when running. A shot involves running the machine and creating a new plasma.
- Every shot, various diagnostics are used to collect different plasma parameters, which are necessary to understand the results of the experiment. Some of these are direct measurements, such as plasma current, and others are derived from the known quantities, such as electron density and temperature.
- After the diagnostics collect the data, that data is digitized if necessary, and then stored as a signal built from the numbers, in a database. That database is called MDSplus.
- MDSplus is a free database—meaning that there are levels. The bottom level with the actual data is called a node.
- Figure 3 is a good example. The top level is LTX. Inside LTX are the trees for each shot (shotnumber). Inside each shot are different categories of diagnostics, in this case magnets. Inside magnets are the different kinds of diagnostic diagnostics, like Mirnov Coils (used to measure magnetic field strength at a point).
- Inside the Mirnov diagnostics is the Mirnov we are looking for, which itself has several attributes. The picture shows the first part of this.
- This is similar to a computer directory (aka Folder) on a computer. The pathname to raw data from a Mirnov Coil would be: LTX/shotnumber/Diagnostics/Magnetics/Mirnov013/shotnumber/.../Raw_Data

**Definitions**

- **Model**: The source for the input page. Written as a Python class, in this application it provides the HTML input form in which the user writes the shotnumber and node they want. The input template renders the model as HTML for your browser to show when you first go to the site.
- **Controller**: The part of the Application that actually contains the commands to show the input template (the model); and calls the compute method. It then saves the plot output from the compute method, and adds that to the output template. This is the app the server runs.
- **Compute**: Accepts the user parameters input to it by the Controller method (from the input form), and uses them to output a plot. For LTX, this is a script which connects to the LTX tree and plots the data from the desired node to the template.

The photos below are from a different project, but the projects are conceptually the same.

**LTX Web Tools**

- **MDplus has an object based interface**.
- **At PPPL, the most common languages are MATLAB, IDL, and Python. Python is considered the best language to learn now.**
- A Python example of an MDSplus script is pictured here.

**Plotting LTX Data**

**MDplus Plotting And Other Uses With Scripts**

- The best way to access MDplus is through programming scripts, using them to plot, save, or print the data from a node.
- Most popular programming languages can be used – MatLab, and IDL are two popular ones. Object Oriented Languages like Python, Java, and C are especially useful. MDplus has an object based interface.
- At PPPL, the most common languages are IDL, MATLAB, and Python. Python is considered the best language to learn now.
- A Python example of an MDSplus script is pictured here.

**CONTACTS**

- **NSTX Web Tools site**: http://nits.pppl.gov/nits/Software/WebTools/index.html
- **Atmosphere, Installation**: Greg Thringwaras
- **Contact**: Nathaniel Sokolow – nsokolow@pppl.gov

**Acknowledgments**

This was a project made possible by funding from the Department of Energy for the Summer Undergraduate Laboratory Internship (SULI) program. This work is supported by the U.S. Department of Energy, Office of Science, Office of Fusion Energy Sciences under Contract DE-AC02-09CH11466. Kyoto University and Lawrence Livermore National Laboratory are gratefully acknowledged for contributing their contributions to this project.

This summer, development commenced on a suite of web tools (basically web applications) which could be used to plot MDSplus data. Hopefully, in the future the scope of the project will expand beyond just plotting, and may eventually be used on NSTX.