

Web Interfaces to MDSplus Data^{*}

W. Davis, P. Roney, T. Carroll, T. Gibney, D. Mastrovito
Princeton Plasma Physics Laboratory, P.O. Box 451, Princeton, NJ 08543-0451

Abstract-- The MDSplus Data Acquisition System has been used successfully on NSTX at PPPL since early 1999. While the X-window-based tools supplied with MDSplus are powerful, and both the IDL programming language and the supplied TDI language are effective for complex and specialized analyses, some users prefer the convenience of applications that can be run in a Web browser. Web users don't need to know about X-windows, or even have accounts or privileges on server computers. The MDSplots page reads MDSplus "Scope" setup files and plots MDSplus signals with overlays. It uses a Perl script to open a pipe to an IDL subprocess. The TreeSearch page provides several options for finding information in MDSplus trees. The MDSshots page lists shots where the maximum for a specified signal is above or below a specified value. A variety of information in an MS-SQL database is also accessible from a Web browser. These utilities could be used at other sites which keep data in MDSplus.

I. BACKGROUND

The National Spherical Torus Experiment [1] (NSTX) experiments at the Princeton Plasma Physics Laboratory (PPPL) has been using MDSplus for fusion since it began operation in February of 1999 [2]. During an experimental cycle (called a "shot") a plasma is produced and around 80 megabytes of data are acquired from instruments in dozens of subsystems and transferred to a central computer system during the subsequent 2-3 minutes [3].

MDSplus [4] is a data acquisition and storage system originally developed for C-MOD, RFX and ZTH, and is now used at 26 facilities world wide. By storing all the data in MDSplus, standard tools can access any type of data with a minimum of specialized knowledge. MDSplus is installed with several X-window/Motif tools, including Traverser for traversing the data hierarchy and setting hardware parameters, and Scope, for plotting x-y traces. The supplied Tree Data Interface Language (TDI) [5] and the Tree Command Language (TCL) also provide low-level capabilities. Remote computers may write to and read from MDSplus servers via

TCP/IP connections from languages such as IDL [6], C++, Visual Basic, FORTRAN and LabView.

NSTX has a heterogeneous group of users. Some prefer the Unix operating system, some VMS, and some Windows. Some prefer Macintoshes as their workstation, some PC's, and others X-terminals. Some researchers can program effectively in a high-level computer language and others cannot. The majority of the scientists on NSTX are not full time on NSTX.

II. WHY USE THE WEB TO ACCESS DATA?

The X-window/Motif tools supplied with MDSplus, and other tools written at various fusion labs [7], [8], are powerful and reasonably flexible. However, an X-windows user needs to know many details such as operating-specific command- and file-syntax, directory names and environmental variables. This can confuse a user unfamiliar with the operating system of the data-serving computer. Additionally, the use of X-windows may require setting additional parameters, such as the TCP/IP address of the X-window "DISPLAY," number of colors, security settings, etc. Accessing data from a Web browser requires much less knowledge in general.

World Wide Web browsers, such as Internet Explorer or Netscape, are familiar to today's researchers. While there can be installation-specific incompatibilities when using a browser, such as special fonts, plug-ins, and screen resolution, basic functions such as making menu selections, entering data into fields, and the display of images, will operate similarly on most computers, regardless of operating system. Web browser access allows, but does not require, the use of a username and password.

Because of the ease-of-use of these tools, there is a growing and appreciative user base at PPPL.

III. OVERVIEW OF WEB TOOLS USED ON NSTX

The Web-based tools for NSTX may be accessed from <http://nstx.pppl.gov/nstx/Software/WebTools/>.

- A. Treesearch - Searches MDSplus node names or their contents for text strings.
- B. Logbook - standard and custom SQL queries to find shots by date, database entries by user, etc.
- C. MDSshotSearch - searches signals within a range of shot numbers for data within user-specified limits.

^{*} This work was supported by DOE Contract DE-AC02-CHO3073.

Lists shot numbers found and the minimum and maximum of the signal.

D. MDSplots – plots MDSplus signals with options to use “Scope” files as inputs and to overlay shots (see Fig. 1).

NSTX MDSplus Plotting Tool

Enter the following information and click on Send:

Shot Number: (default is the current shot-1)
(search for desired shot numbers)

Overlay Shots: (optional)

Select Scope File Source:

Choose From List, OR Enter Scope filename in /p/nstxusr1/util/scopes/

Coil Currents
Density
ECH
Mimov
NB Summary
Neutrons
NPA_Compare_12
RF Power
Small Summary
Wall Conditioning
Waveform Summary
EBW

Plot Ranges:

X: From Scope from to (sec)

Y: From Scope Autoscale

Plot Labels:

From Scope (no shot #) From MDSplus
 Tag Names

You may wish to read about the [details on this page](#).

Other Optional Inputs:

Signal Units Displayed: None on Y-axis append to Title

Median-Smoothing Neighborhood: (Defaults to no smoothing)

Layout of Plots: # of rows: # of columns: (Blanks are OK)

Size of Plot Window: Horizontal: Vertical: (pixels)

Fig. 1 Web page for MDSplots. Users fill in fields and make selections with the mouse to see plots of experimental data in any Web browser.

- E. MDSlist – like MDSplots, but a table of numbers is displayed, suitable for importing into a spreadsheet.
- F. Web Traverser – Traverses an MDSplus data tree hierarchy and plots signals (written in Java) [9].
- G. Dbaccess – View and plot information from MS-SQL databases derived from MDSplus data.

H. Shot Listing – lists the shots associated with particular experimental proposals (XPs), and lists shot ranges for each run day.

Some of the most frequently asked questions about the NSTX data stored in MDSplus involve determining the node names for signals of interest and determining shots that satisfy certain criteria. SignalLabels, TreeSearch, Logbook,

MDSshotSearch, the Web Traverser, and Dbaccess.asp make it easy to find signal names and shots of interest.

Both the Logbook and DBaccess application query an MS-SQL database containing experimental parameters and comments for each shot. Dbaccess uses an active server page (ASP) to generate up-to-date lists of tables and data items.

The MDSplots, TreeSearch, MDSshots, and MDSlist are written in simple HTML, but connect through TCP/IP sockets to Perl [10] scripts that in turn make IDL Remote Procedure Calls (RPC's). MDSplots allows MDSplus "Scope" setup files as an input, or accepts individual MDSplus node names, and allows the user to overlay data from several shots. TreeSearch searches MDSplus nodes, with various options, and displays node and tag names that satisfy the search criteria. MDSshots lists shots that satisfy certain criteria, such as those with greater than, or less than, a certain value for common parameters, or from user-specified MDSplus node names.

IV. Design Details of the MDSplots Web tool

The data flow of the MDSplots Web Tool is shown in Fig. 2.

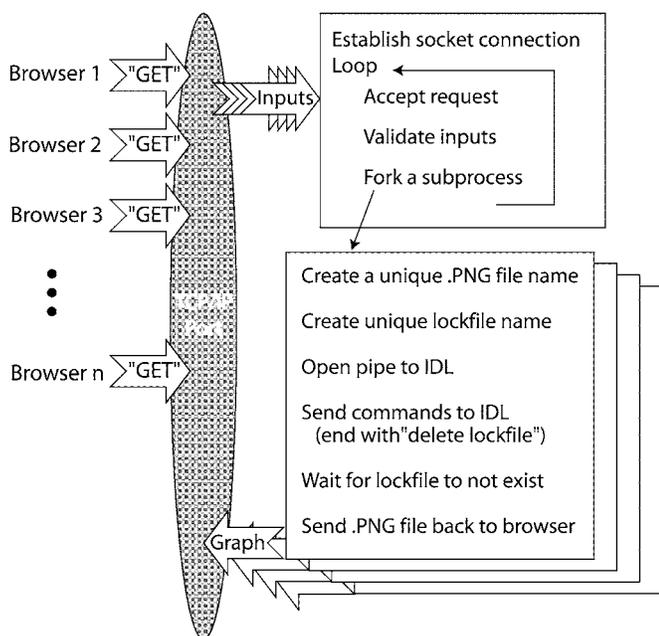


Fig. 2 Data flow of the MDSplots page and the Perl server.

The input fields on the Web page are part of an HTML form. When the user hits the "Send" button on the Web page, field names and inputted data are sent to the host computer. A Perl script, whose functions are listed in the boxes in Fig. 2, needs to be running on the host computer, and have made a connection to the correct socket. The Perl script then stays in a loop with a limited number of commands including "forking" a separate subprocess for each set of inputs. The commands in each forked process may take several minutes, but inputs from other Web browsers can be processed concurrently by the main loop. Within a forked process, a unique portable network graphics (PNG) file name is created (based on the PID of the

process), and another unique file name to act as a lock. Commands are piped to a newly-spawned IDL process which end with an IDL command to delete the lockfile. The Perl script then waits for the lockfile to be deleted by the IDL process after it has created the PNG file. When the lockfile ceases to exist, the Perl script sends the newly-created PNG file back to the Web browser that initiated the request.

V. SECURITY CONCERNS FOR WEB APPLICATIONS

The NCSA Web page entitled "Writing secure CGI scripts" [11] states:

Any time that a program is interacting with a networked client, there is the possibility of that client attacking the program to gain unauthorized access. Even the most innocent looking script can be very dangerous to the integrity of your system.

Two distinct security concerns require different policies and practices. Of primary importance is preventing unauthorized access to system functions. These include access to passwords and other private data and the ability to modify or destroy experimental data or source code. Protecting experimental data from unauthorized or incorrect use is a secondary concern. While the only way to be sure a computer is secure is to make it inaccessible from the outside, security concerns can be managed by following established practices.

The flexibility of Perl allows for the execution of operating system commands, such as those for deleting files. The `-T` switch in Perl enables "taint" checking. The basic principle of taint checking is "to mark as tainted anything that comes from outside the program, including command line arguments, environmental variables and file input. Tainted data may not be used directly or indirectly in any operation that invokes a subshell, or in any operation that modifies files, directories, or processes" [10].

The Perl script we use passes commands based on user inputs to an IDL server. These commands can include filenames entered on the Web page. This necessitates additional checks. The Perl script explicitly checks all input strings and eliminates characters that would allow the Perl script or the IDL process to access the Unix shell or to execute other dangerous commands. The filenames entered are not displayed or executed; they are only read by specific IDL code.

"Buffer overflows" were by far the most frequent form of security vulnerability for remotely accessible networks in the 1990's [12]. An example of this is when program inputs exceed the input buffer sizes of the program and the remaining characters somehow become executable code. The Web applications described here were tested with a variety of large buffer sizes, and no vulnerabilities or unexpected behavior were evident. IDL always does explicit bounds checking, unlike the C programming language. If the scripts supporting

our Web applications were compromised, any damage would be contained, since they do not execute with system privileges.

PPPL has a firewall between its internal computers and the Internet. By default, a user must have a SecurID Badge, and enter a username, Secure ID Pin and a number from the badge (which changes every minute). Some of our Web pages require this form of authentication and some are open to anyone.

VI. SUMMARY

The use of Web browsers is well understood by most computer users. The Web tools described here for locating and displaying MDSplus data were developed with no more effort than traditional X-windows/Motif applications and have increased accessibility to the data for many researchers. The Web tools that do not use Java do not have some of the features of the corresponding X-windows tools, such as using the mouse for panning and zooming, but require the users to know far fewer details. Security concerns are more complex with Web applications, but by following some relatively simple established practices, we feel we have struck a good balance between security and accessibility.

ACKNOWLEDGMENT

The authors thank Tom Fredian and Josh Stillerman from MIT for support with MDSplus issues, Stan Kaye and Dennis Mueller for guidance on user requirements for NSTX, and Steve Davis with guidance on security.

REFERENCES

- [1] Kaye et al., The physics design of the National Spherical Torus Experiment. *Fusion Technology*, **36**, July 1999, p. 16, or <http://nstx.pppl.gov/index.shtml>.
- [2] W. Davis, P. Roney, T. Carroll, T. Gibney, D. Mastrovito, "The Use of MDSplus on NSTX at PPPL." Presented at the 3rd IAEA TCM on Control, Data Acquisition, and Remote Participation for Fusion Research, Padova, Italy (2001), *Fusion Eng. Des.*, in press.
- [3] P. Roney, K. Tindall, S. Davis, "Computer and Network Infrastructure of NSTX" these proceedings.
- [4] J.A. Stillerman, T.W. Fredian, K.A. Klare, G. Manduchi, MDSplus Data Acquisition System. *Review of Scientific Instruments*, **68** (1) January, 1997, p. 939.
- [5] Tree Data Interface (TDI) Language.
http://www.mdsplus.org/tdi/tdi_tutorial
- [6] IDL – The Interactive Data Language
<http://www.rsinc.com/idl/>
- [7] J. Schachter, Q. Peng, D.P. Schissel, "Data analysis software tools for enhanced collaboration at the DIII-D National Fusion Facility," *Fusion Engineering Design* **48** (2000) 91-98.
- [8] FAS User's Guide,
<http://nstx.pppl.gov/nstx/Software/fas/index> page.
- [9] G. Manduchi, C. Taliercio, A. Luchetta, "The Java interface of MDSplus: towards a unified approach for local and remote data access," *Fusion Engineering Design* **48** (2000) 163-170.
- [10] L. Wall, T. Christiansen, and J. Orwant, *Programming Perl*, 3rd ed. Sebastopol, CA: O'Reilly, 2000, pp. 559.
- [11] Writing secure CGI scripts.
http://hoofoo.ncsa.uiuc.edu/cgi/security_page.
- [12] C. Cowan, P. Wagle, C. Pu, S. Beattie, J. Walpole, "Buffer overflows: Attacks and Defenses for the Vulnerability of the Decade" Proceedings of DARPA Information Survivability Conference and Expo (DISCEX), <http://schafercorp-ballston.com/discex> To appear at SANS 2000, <http://www.sans.org/newlook/events/sans2000.htm>. For a copy, see http://immunix.org/documentation_page.