

The Use of MDSplus on NSTX at PPPL

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 since the 1999 startup of the National Spherical Torus Experiment (NSTX) for control, data acquisition and analysis for diagnostic subsystems. Prior to first plasma on NSTX, software engineers collectively spent 1 man-year to learn the system and bring it to operation, and 4 full-time equivalents continue to support its uses. The 30 primary, on-site physicists and 25 part-time physicists adapted to the MDSplus software tools with no real difficulty. After a few uses, most users valued the features and usability of Scope and Traverser, the most commonly used MDSplus components. Within 3 minutes of the 1-second plasma “shot” on NSTX, 40 MBs are acquired from CAMAC crates on a VMS system, 22 MBs from UNIX systems, and 13 MBs from PC systems, and loaded into MDSplus hierarchical data structures, called trees. Local users and remote collaborators alike configure their X-windows to display dozens of plots, as data becomes available. Data can be read from MDSplus trees into high-level language codes at 500 KB/s on our VMS systems, and about 20% slower on our Unix systems. All NSTX data stored in MDSplus is always on-line from VMS-served RAID disks. Information is written directly from PC-based “turn-key” systems into MDSplus, using routines called from C++, LabView, Visual Basic and IDL. Several legacy FORTRAN analysis codes have been converted to use MDSplus with minimal effort. Experiment-specific tools and routines in our user-written IDL libraries have been layered on top of MDSplus. Database tools are being converted from INGRES to MS-SQL. Web-based plotting, tree-searching, and database tools have also been developed and various parts of MDSplus have been documented on the web. Many of these tools should be useful to other MDSplus sites. The support from the developers at MIT, especially with the intricacies of the TCL and TDI languages, was timely and insightful. The use of MDSplus has resulted in both a better system and a significant cost savings for NSTX.

Keywords: Data acquisition; Data visualization; Data management; MDSplus; NSTX

1. Background

The National Spherical Torus Experiment (Kaye, et al. [1]) (NSTX) began fusion experiments at the U.S. Department of Energy’s Princeton Plasma Physics Laboratory [2] (PPPL) in February of 1999. NSTX is a proof-of-principle experiment predicted to have more efficient magnetic confinement and better stability than conventional tokamaks. During an experimental cycle, called a shot, a plasma is produced and 10’s of megabytes of data are acquired from instruments in dozens of subsystems and stored in local memories. Data is then transferred to a central computer system during the following 2-3 minutes. Between shots, there typically are 30-40 researchers examining plots of raw and analyzed data to evaluate the particular experiment.

MDSplus (Stillerman, et al. [3]), developed by the Massachusetts Institute of Technology (MIT), the RFX group at CNR in Padua, Italy, and the ZTH group at Los Alamos National Laboratory, is a data acquisition and storage system which allows an experiment to be completely described within a hierarchical data structure, and which includes a variety of X-window/Motif applications for hardware setup (e.g., Traverser [4]) and data display (e.g., Scope [5]). The majority of the current diagnostics on NSTX follow the “standard” MDSplus model, accessing CAMAC from a VMS computer, but the “national” aspect of NSTX requires that many pre-assembled diagnostics from collaborators be integrated, as well. These typically come on “turn-key” PC systems which are written in C++, Visual Basic, or LabView [6]. Additionally, data is stored from an Experimental Physics and Industrial Control System (Sichta, et al. [7]) (EPICS) and from a

* Corresponding author. E-mail: bdavis@pppl.gov, Phone: (609)-243-2546, Fax: (609)-243-3086

VME-based real-time feedback control computer (Gates, et al. [8]). By storing all the data in MDSplus, standard tools can access any type of data with a minimum of specialized knowledge.

NSTX has an especially 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 can not. Scientists who participate in NSTX include 53 from PPPL and 80 from 17 US institutions and other countries including Japan, Russia, Korea, Ukraine and Canada. The majority of these participants are not full time on NSTX. This makes ease-of-use an essential feature.

2. Reasons for Choosing MDSplus

Since it is usually less expensive to adapt existing software than to write it from scratch, we evaluated possible candidates. The costs and benefits of converting software written for previous PPPL experiments were compared with estimates of adapting external packages. Whatever existing software was chosen would need to be easily customized for our site-specific requirements. The selection of a software system needed to take into account the relatively small support staff and part-time nature of many of the users, as well as the choice to make use of our \$6 million inventory of legacy CAMAC data acquisition and control modules. EPICS was chosen for NSTX's Distributed Control System, and an analogous system was needed for the shot cycle requirements.

MDSplus was designed for shot cycle-based experiments similar to NSTX. EPICS systems, on the other hand, are generally used for experiments like large accelerators and telescopes. Events in a typical NSTX shot cycle are listed in Table 1.

Time (min:sec)	Action
Earlier	Users Adjust Settings
-2:20	Checks and Ramp-ups
-1:00	Files are Created
0:00	<i>Plasma Shot Begins</i>
0:01	Shot Finishes
0:09	Data Acquisition Begins
2:30	Data Acquisition Finishes
After	Users Analyze Data

Table 1. NSTX Shot Cycle events during a typical 5-10 minute period

The requirements for different phases of the shot cycle were well supported by MDSplus.

MDSplus supported most of the existing CAMAC modules we planned to use on NSTX. We would be able to add code for other modules we wished to support.

The predecessor to MDSplus, the Model Data System (MDS), was used successfully on PLT fusion experiments at PPPL in the mid-1980's. Users were pleased with the forms-based setup procedures, consistent data access and easy-to-use plotting tools. Since MDSplus has been used at many fusion research facilities throughout the world, it seemed unlikely that we would encounter intractable problems.

The Interactive Data Language (IDL) [9] was introduced to PPPL with MDS and has since become popular among many users. The basic MDSplus interface calls are provided for IDL, and there are many IDL codes within the Fusion community that access MDSplus trees.

These reasons gave us confidence that we could use MDSplus for our diagnostic data acquisition and storage on NSTX.

3. NSTX Computer Configuration

As seen in Fig.1, a variety of computers are used at PPPL for NSTX.

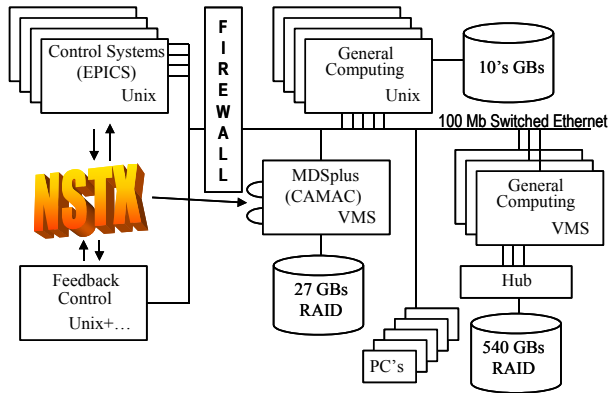


Fig. 1. NSTX Computer Schematic

In the middle of the picture, a locally-written program receives notice of shot cycle events and invokes various components of MDSplus. All access to the diagnostic CAMAC highways is done on this VMS computer. Current shot data is written to a local RAID set, and is kept there for one week, after which time it is moved to the larger file systems shown in the lower right. A design goal for NSTX is to keep all data on magnetic disk. On the left side of Fig 1, more sensitive applications, such as those controlling power supplies and vacuum valves, are run from computers behind a firewall. The Control Systems use EPICS and the real-time Feedback Control system acquires data in VME modules. Data from these systems and from the PC's (Mastrovito [10]) are stored into MDSplus trees after the shot. All but the EPICS systems use calls over TCP/IP to load their data into the trees.

The models of the major computers from Fig. 1 are listed in Table 2.

OS	Model	# CPU	MHz	MBs
VMS	DS20E	2	667	3500
VMS	AlphaServer 800	1	500	1000
VMS	Alpha 2100A 4/275	3	275	768
VMS	Alpha 2100 4/275	3	275	500
Solaris	UltraSPARC-II	4	400	2000
Solaris	UltraSPARC-II	2	400	1500
Solaris	(2) Sparc Ultra 30	1	300	768

Table 2. The Major PPPL Computers used on NSTX.

Users displaying and analyzing NSTX data typically run on Unix or VMS systems shared with other general users at PPPL. The MDSplus utilities, and routines written in IDL, look and feel the same on both platforms. PC's and, potentially, Macintosh computers, can also bring data from the MDSplus trees directly into applications written in languages that include IDL, LabView, Visual Basic, FORTRAN, C, and C++.

Within 3 minutes of the 1-second plasma "shot" on NSTX, 40 MBs are acquired from CAMAC crates on a VMS system, 22 MBs from UNIX systems, and 13 MBs from PC systems, and loaded into MDSplus hierarchical data structures, called trees.

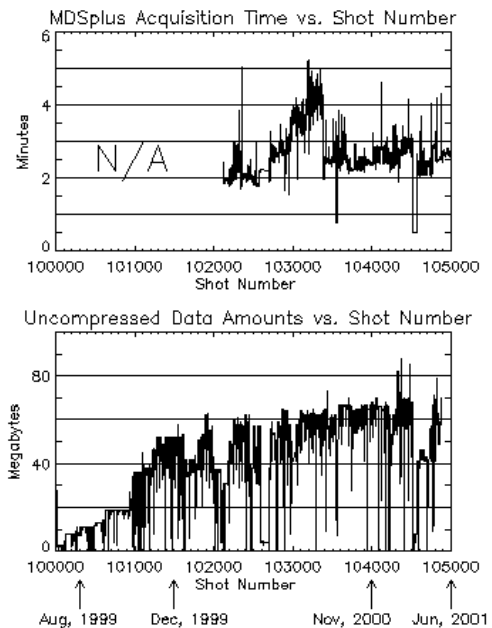


Fig. 2 Data Acquisition Statistics for NSTX

As shown in Fig. 2, uncompressed data amounts per shot have not been increasing rapidly, but this could change dramatically, say by a factor of two, at any time, if various analysis programs are run for more cases or with higher time resolution. The sudden drop in acquisition times around shot 103400 corresponds to the addition of a second CAMAC highway. Raw data is stored in MDSplus trees with compression factors of 2.5-3. Analyzed data, which is usually floating point data, does not compress significantly.

Local users and remote collaborators configure their software to display dozens of plots on their X-windows as data becomes available. Data can be read from MDSplus trees into high-level language codes at 500 KB/s on our general-use VMS systems, and about 20% slower on our Unix systems. All NSTX data stored in MDSplus is always on-line from VMS-served RAID disks.

4. Labor and Learning Costs

While a basic MDSplus system can be brought up in a matter of weeks, local adaptations and extensions can take a widely varying amount of time. Since there are so many differences from one site to another, not to mention the differences in the background of the users and implementers, the labor costs for implementing MDSplus on NSTX may have very little relevance to other sites. Nevertheless, we include our experience here as a possible rough guide to others.

MDSplus documentation is a work in progress, and is an acknowledged shortcoming at this time. It was sometimes helpful to examine MDSplus source code. The Tree Data Interface (TDI) language [11] and Tree Control Language [12] (TCL) are powerful tools included with MDSplus, but are documented only in a reference format. The best way to see how these are used is to find existing examples from other users. Debugging TDI programs is made more time consuming by the lack of a debugger.

Prior to first plasma on NSTX, software engineers collectively worked 2.5 man-years in preparation, but less than half of this was involved with MDSplus. Software engineers spent several man months learning various components of MDSplus in depth, and, of course, much more learning occurred after operations began. A typical end user might just learn how to change a few data acquisition parameters in the Traverser tool and look at the data with a Scope set up by someone else. Users can reach this level of proficiency in a few hours, if there is a basic understanding of the computer system and X-windows. At the other extreme, a user may wish to develop complex algorithms in TDI, or in another language such as IDL, and this might take weeks or months to learn and perfect. The 30 primary, on-site physicists and 25 part-time physicists adapted to

the MDSplus software tools with no real difficulty, but instructing and assisting these users typically keeps a programmer busy one-quarter to one-half time during experimental operations.

It might take an experienced person a day to layout and create the initial data tree structure for a diagnostic. For the "Operations" tree of NSTX, which has 200 signals with different calibration algorithms, etc., it took weeks to write TCL code to generate the tree from a spreadsheet. To design and implement the data management system for NSTX data (file locations, archiving, etc.), took about a man-month. Four programmers (full time equivalents) continue to support the use of MDSplus on NSTX; this includes assistance using the tools supplied with MDSplus as well as writing customized data acquisition setup and analysis programs.

Most of the 30 different types of CAMAC modules we have used on NSTX were well supported by the standard MDSplus distribution. We wrote or modified MDSplus acquisition modules for half-a-dozen CAMAC modules. The first one took a man-month because the programmer only had modest experience with C and Motif. Subsequent modules took much less time.

5. Locally-developed Tools

Several legacy FORTRAN analysis codes have been converted to use MDSplus with minimal effort. Experiment-specific tools and routines in our user-written IDL libraries have been layered on top of MDSplus, and simplify the writing of specialized IDL codes. Data is automatically loaded into several MS-SQL databases between shots. Database tools are being converted from INGRES to MS-SQL.

The X-windows-based program called Scope is supplied with MDSplus, and is powerful and efficient for any number of single x-y traces per frame (the more fully-featured Java version (Manduci, et al. [13]) is only used by a few NSTX collaborators on PC's). Some users need more elaborate or customized programs for particular applications. The features most requested over those of the original Scope program are the ability to overlay plots, plot in color, and plot two-dimensional data. Some users write their own display programs in IDL,

but these often have limitations, such as color incompatibility with other applications, the inability to print easily, non-resizable graphics, bombs when used in unexpected ways, etc. We have tried to bring the General Atomics ReviewPlus (Schachter, et al. [14]) plotting tool into use at PPPL, but difficulties remain. Two other programs have evolved at PPPL for the generalized plotting of MDSplus data, but we are still looking for the "ultimate" general plotting program. The FAS [15] (Fusion Analysis Software) was developed at MAST [16] and has some attractive features, like a GUI polynomial expression generator, well-documented ways to link in customized IDL routines, and several years of stability to reduce bugs. MDSw [17] is another program used by around 10 NSTX researchers. It grew out of the need for a template widget plotting program for MDSplus signals and includes many commonly desired features, such as resizable graphics windows, saving to a variety of file formats, point-and-click interfaces to most IDL system and axis parameters, the use of the mouse for zooming, wild-card searches for MDSplus signal names, and the ability to save one's settings to a file for later use.

Some users are willing to give up some of the features and performance of the MDSplus tools and IDL codes for simpler point-and-click web pages to access their data. When they use the web, they don't need accounts, passwords, x-windows, knowledge of directories, color settings, etc. A few web-based utilities have been developed [18] at PPPL and have a rapidly growing and appreciative user base. These web tools are written in simple HTML, but connect to more sophisticated applications on the web server. The MDSplot, TreeSearch, and MDSshots utilities connect through TCP/IP sockets to Perl [19] scripts which in turn make IDL Remote Procedure Calls (RPC's). MDSplot.html reads scope files as an input, or accepts individual MDSplus node names, and allows the user to overlay data from several shots. TreeSearch.html searches MDSplus nodes, with various options, and displays node and tag names which satisfy the search criteria. MDSshots.html (see Fig. 3) lists shots which 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.

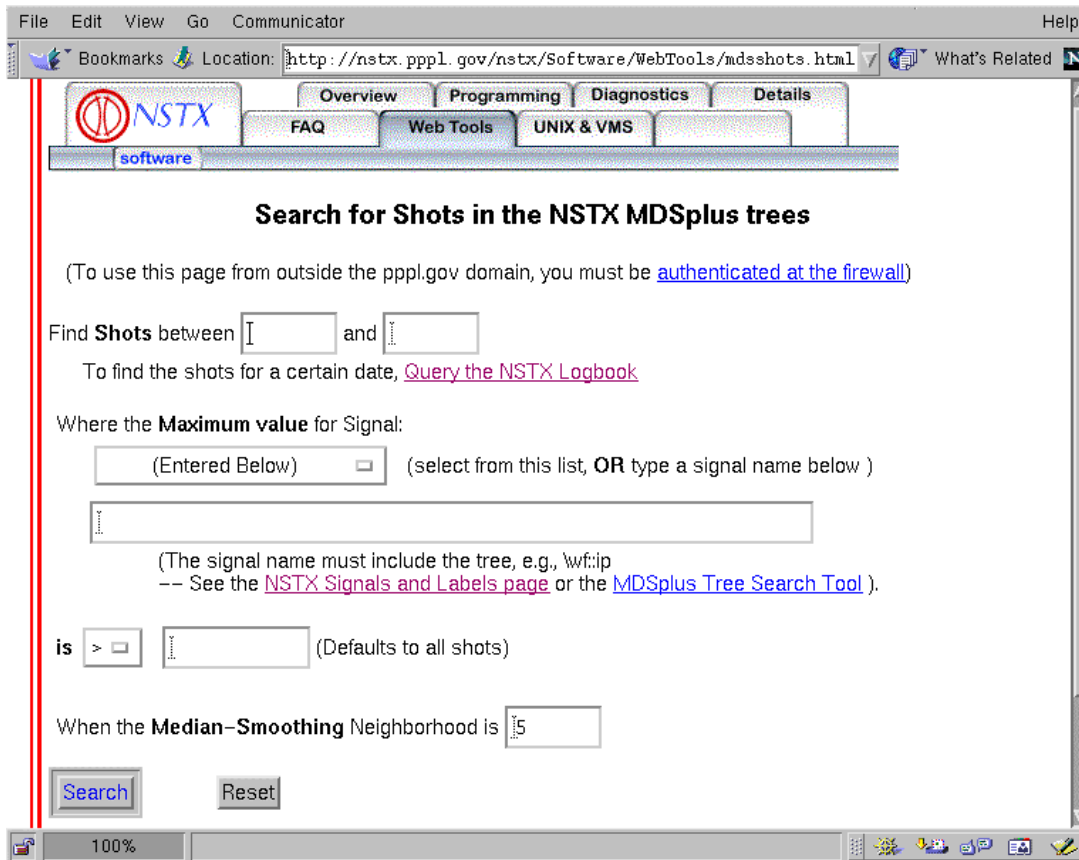


Fig. 3. NSTX Shot Search Web tool.

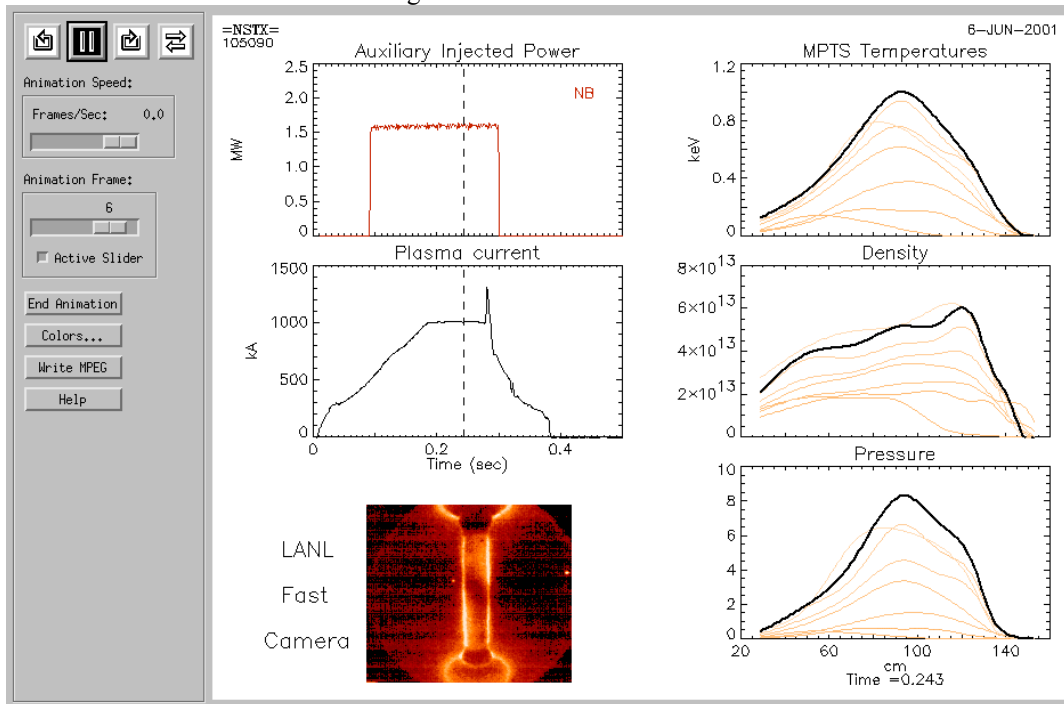


Fig. 4. IDL Widget for animating various types of MDSplus data.

The tool shown in Fig. 4 allows researchers to understand this data much more easily and

illustrates the power of using MDSplus and IDL for developing custom analysis programs. The common interface to data through MDSplus calls, and the availability of many IDL codes, facilitated the writing of this utility in a single day. Plots on the right are profiles, with the minor radius of the plasma as their horizontal axis, while the plots on the left are versus time. Images from the LANL Fast Camera are animated in the lower left. The profile at a particular time is drawn in black, while others are drawn in a lighter color. As the VCR-like controls advance the animation, the dashed lines on the plots on the left move to correspond to that time in the trace.

Our World Wide Web documentation on using MDSplus on NSTX has been evolving. The tabs shown in Fig. 3 allow users to quickly find the page most likely to meet their need. The FAQ (Frequently Asked Questions) page, although incomplete, has been particularly useful. Similarly, by organizing user-written general IDL routines [20] by category (automatically), programmers are much more likely to find an existing routine. There are IDL tutorials for beginning IDL programmers and for more advanced plotting. The CAMAC Control Widget Examples [21] listed on our web site may also be of use to users at other sites.

6. Next Steps

We hope to extend our Web documentation this year, particularly the FAQ, and implement a good search facility.

We plan to organize informational messages better, such as relevant CAMAC or timing errors, and provide automatic notification (and the ability to turn it off!). Several automatic programs run after the shot to analyze and display data, but errors are only written to a log file, and often go unnoticed. While MDSplus events are a powerful and convenient way to notify software tasks, it is not always evident which events are being declared, and for what purpose. It would help us diagnose some problems if the time and source of event declaration were easily available.

In general, additional diagnostics on NSTX will not be CAMAC-based, where data rates are megabytes per minute, but will probably be PCI-based, where data rates are megabytes per

second. Ideally, the addition of new diagnostic data will be as easy as it is now to add CAMAC data using supported modules.

A few NSTX users have specialized spectral analysis codes, written in IDL or FORTRAN, which access MDSplus data. In the next 6-12 months we plan to generalize these tools for easier use by others.

7. Summary

MDSplus has been used successfully on NSTX for diagnostic data acquisition and the storage of all NSTX data. The turn-key aspects of MDSplus were found to be logical and reliable. Adding new features, such as support for new CAMAC modules, or including data from other systems was more difficult, but not unreasonably so. The support from the developers at MIT, especially with the intricacies of TCL and TDI, has been timely and insightful. The use of MDSplus on NSTX has saved man-years of labor, and has provided a sound and powerful system which can grow with our requirements, and those of the fusion community.

Acknowledgements

The authors thank G. Oliaro and S. Davis for guidance in designing the use of MDSplus on NSTX, and P. Sichta for work on communication with the EPICS system. Many thanks to T. Fredian and J. Stillerman from MIT for patient help with MDSplus issues. We also thank S. Kaye, D. Mueller, and R. Budny for reviewing this material. This work was supported by DOE Contract DE-AC02-CHO3073.

References

- [1] Kaye et al., *Fusion Technology*, **36**, July 1999, p. 16, or <http://nstx.pppl.gov/index.shtml>
- [2] The Princeton Plasma Physics Lab, <http://www.pppl.gov>
- [3] 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.
- [4] MDSplus Traverser, <http://nstx.pppl.gov/nstx/Software/TRAVERSER.html>
- [5] MDSplus Scope Plotting Utility, <http://nstx.pppl.gov/nstx/Software/SCOPE.html>

[6] LabView, <http://www.ni.com/labview/what.htm>.

[7] P. Sichta, J. Dong, Startup of the Experimental Physics Industrial Control System at NSTX, *18th Symposium on Fusion Engineering*, Albuquerque, NM (1999).

[8] D. A. Gates, D. Mueller, C. Neumeyer and J. R. Ferron, Control System Development plan for the NSTX, IEEE RT99 Conference (June, 1999), *IEEE Trans. Nucl. Sci.*, 1999.

[9] Research System's Software <http://www.rsinc.com/>

[10] D. Mastrovito, Integration of Windows Applications with MDSplus Data Acquisition On NSTX 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.

[11] TDI (Tree Data Interface) Information, http://nstx.pppl.gov/nstx/Software/TDI/TDI_TOP.HTML

[12] MDSplus Tree Command Language (TCL) command descriptions index, http://nstx.pppl.gov/nstx/Software/TCL_index.html

[13] 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.

[14] 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.

[15] FAS Users's Guide, <http://nstx.pppl.gov/nstx/Software/fas/index.html>

[16] UKAEA Fusion: MAST - Mega-Amp Spherical Tokamak, <http://www.fusion.org.uk/mast/index.html>

[17] MDSw – An IDL widget for plotting MDSplus data, http://w3.pppl.gov/~bdavis/pppl_idl_routines.html

[18] Web-based tools for NSTX, <http://nstx.pppl.gov/nstx/Software/WebTools/>

[19] Perl, <http://www.perl.org/>

[20] IDL Pages at PPPL, <http://nstx.pppl.gov/nstx/Software/IDL/idl.html>

[21] IDL CAMAC Control Widget Examples, <http://w3.pppl.gov/~bdavis/camac/camacwidgets.html>