

PSFC/JA-97-17

**The MDSplus Data Acquisition System,  
Current Status and Future Directions**

J.A. Stillerman, T.W. Fredian

July, 1997

Plasma Science and Fusion Center  
Massachusetts Institute of Technology  
Cambridge, MA 02139

To be published in proceedings of IAEA Technical Committee Meeting on Data Acquisition and Management for Fusion Research.

This work was supported in part by the U. S. Department of Energy Contract No. DE-AC02-78ET51013. Reproduction, translation, publication, use and disposal, in whole or in part by or for the United States government is permitted.

# THE MDSPLUS DATA ACQUISITION SYSTEM, CURRENT STATUS AND FUTURE DIRECTIONS

Joshua Stillerman, Thomas W. Fredian - MIT Plasma Science and Fusion Center

## Abstract

The MDSplus data acquisition system was developed in collaboration with the ZTH Group at Los Alamos National Lab and the RFX Group at CNR in Padua, Italy and is currently in use at MIT, RFX in Padua, and TCV at EPFL in Lausanne. MDSplus is based on a hierarchical experiment description which completely describes the data acquisition and analysis tasks and contains the results from these operations. It also includes a set of X/motif based tools for data acquisition and display, as well as diagnostic configuration and management. These tools were designed to operate in a distributed, client/server environment with multiple concurrent readers and writers to the data store. An interface to a relational database is provided for storage and management of processed data. A commercially available package called IDL is used as the primary data analysis and visualization tool. The current projects include a new interface to the electronic logbook, tools for remote collaborators and WWW access, and a port of the system to UNIX and Windows-NT/95.

## 1. Background

The MDSplus data system was written by us in collaboration with the RFX Group at CNR in Padua, Italy and The Zth Group at Los Alamos National Laboratory.<sup>1</sup> It is currently being used by both Alcator C-Mod and PTFAL at the MIT Plasma Science and Fusion Center, RFX at CNR and TCV at EPFL in Lausanne. MDSplus is also being considered for use at Princeton Plasma Physics Laboratory for the NSTX experiment and for analysis results from TFTR. These experiments represent a broad range of sizes from a few dozen up to thousands of signals. This list includes members of the original software development collaboration and sites which are solely users of the system.

MDSplus has also been used as both a front end and back end processor for physics modeling codes such as TRANSP<sup>2</sup> and EFIT. Over the course of one day, it was installed at Princeton Plasma Physics Laboratory and access to the existing TFTR waveform data was implemented. If it is adopted by other sites it could form the basis for data sharing in the fusion community.

## 2. Data Storage

At the core of MDSplus is the centralized data store which is arranged hierarchically. In this one place, or tree, information about all aspects of the experiment from diagnostic setup to highly processed analysis results, are stored. One simple interface provides users and applications access to all of these types of data. The hierarchy allows the data to be organized so that diagnostics can be self descriptive and the data can retain their meanings longer. This is especially critical as the number of data items increases.

Before the pulse, the dispatcher creates the pulse file and tells server processes to arm the data acquisition hardware with the setup information from the tree. After the pulse the dispatcher directs the servers to acquire the raw data and perform the automatic data analysis. Once this completes the tree for that shot contains:

- all of the setup information for the diagnostics, calibration and geometry values
- setup for the data acquisition and high speed timing
- power supply waveforms and feedback control matrices
- task description, scheduling and status
- raw data from the engineering systems and diagnostics
- processed data from automatic data analysis.

Processed data from interactive analysis is added as it becomes available. One set of tools and one programming interface can be used to access all of these types of data. Read and/or write access to these data is granted to authorized users.

In order to store all of these types of information the tree must be able to contain a wide variety of data types. Nodes in the hierarchy can contain scalars or arrays of the common primitive types including string, byte, word, long, float, double, complex, ... For digitizer channels, task descriptions, and other commonly occurring collections of items, MDSplus implements some special types.

- SIGNAL - used to associate independent axes with arrays. Subscripting can be done on either the array indices or the independent parameter(s).

- WITH\_UNITS - associates a units string with a node in the tree, or a term of one these complex data items. For example, a digitizer channel would have 'Volts', 'Counts' and 'Seconds' associated with its value, raw data, and first dimension, respectively.
- WITH\_ERROR - associates error bars with a data item.
- RANGE - composed of start, end and increment terms. Ranges are used in the representation of clocks. Multifrequency and burst mode clocks are stored as ranges whose terms contain arrays.
- ACTION, TASK, and DISPATCH are used to represent automatic data acquisition and analysis jobs and their scheduling and status.
- EXPRESSION - any node in the tree, or term in one of these special types can contain an expression instead of primitive data type. These expressions are evaluated by the built-in expression evaluator when they are referenced. They are used to represent the logical and physical relationships between items in the tree. This is the same expression evaluator which forms the basis for the user interfaces to the data..

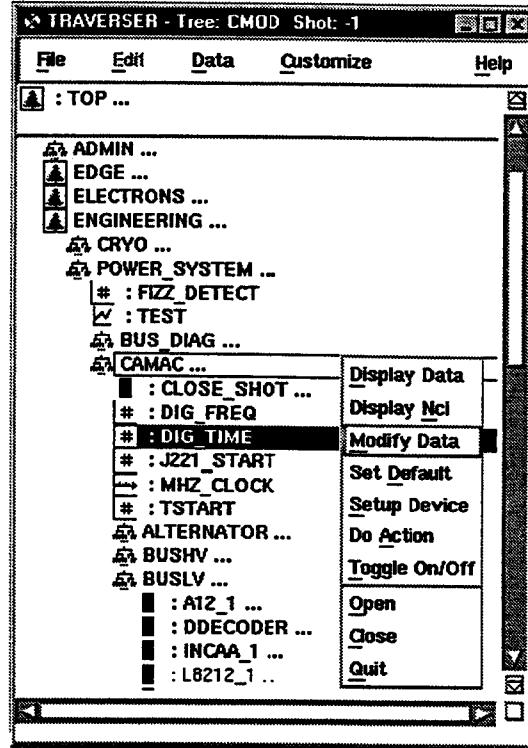


Figure 1 TRAVERSER display

This rich set of types allows the pulse file to contain a fairly complete description of the experiment. A typical diagnostic might have branches associated with data acquisition, data analysis, calibration and geometry, and results. Under each of these, further structure can be employed as needed to keep the data organized.

An X-Windows based application is provided to interactively construct and manipulate the trees. Figure 1 shows a typical screen from this program. The display can be expanded and collapsed to show and hide the nodes' descendants using the mouse. This tool is also used for data acquisition device setup.

The C-Mod plasma control system<sup>3</sup> uses the tree to store all of the power supply waveforms, gains, settings and the actual values which are downloaded to the real time control computer. This package, written in IDL using the standard data access calls, is the physics operator interface to C-Mod. After the operator draws the control waveforms and PID gains, and specifies the control algorithms, matrices and scaled waveforms are computed and written back into the tree. During the initialization phase these outputs are downloaded from the tree into the real time control computer.

### 3. Data Display

DWSCOPE is a X-Motif based application for display of one dimensional data. It presents the user with a table of up to sixty-four panels each containing an X/Y plot. The panels can be set to update automatically when new data becomes available. Graphical pan and zoom as well as tracking crosshairs with numeric readout are supported. During an Alcator C-Mod run, most of the forty workstations and XWINDOWS displays in the control room are running multiple copies of the DWSCOPE with a large number of traces each. Within the first few minutes after a C-Mod pulse, copies of DWSCOPE refresh thousands of panels in the control room and at remote sites. See Figure 2 for a typical DWSCOPE screen.

For each panel the user specifies expressions to evaluate for the X and Y axes. In addition, plot style, scaling, labeling and automatic update behavior is configurable. Copy and paste from and to DWSCOPE panels, as well as from the TRAVERSER, is a convenient mechanism for constructing new setups.

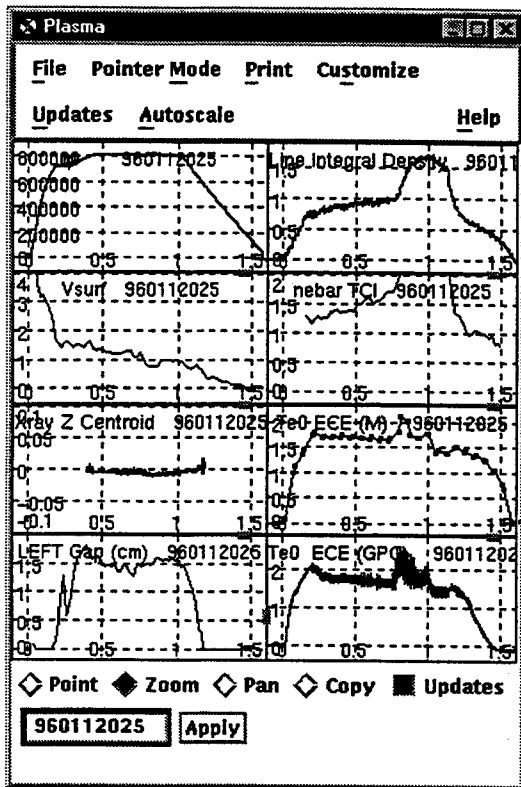


Figure 2 DWSCOPE screen capture

#### 4. Expression Based Data Access

All of the interfaces to the data in MDSplus are based around an expression evaluator. These expressions can be stored in the tree; used as arguments to the user interface routines in C, FORTRAN or IDL<sup>4</sup>; or specified in interactive data analysis and display programs like the DWSCOPE. They are also used to implement various aspects of device and shot cycle support.

To retrieve data from the tree, the user specifies an expression to evaluate. This is usually as simple as reference to a node in a tree or an arithmetic expression of nodes and constants. Type conversion and vector/scalar operations are performed automatically. The language includes hundreds of built-in functions, user function definitions with local and global variables, as well as the ability to call routines in external libraries. These are mostly used to implement system functions.

The IDL and compiled language interfaces to MDSplus consist of just four routines. They are presented here in IDL syntax. Equivalent calls are available in C, C++, and FORTRAN.

1. MDS\$OPEN, tree, shot
2. answer = MDS\$VALUE(expression [args...])
3. MDS\$PUT, node, expression [args...]
4. MDS\$CLOSE.

These four routines provide complete read and write access to the MDSplus data system. The optional arguments to MDS\$VALUE and MDS\$PUT are values to be substituted into the expression when it is evaluated or written. For example to scale a signal by a constant:

```
var = mds$value('\IP*7.8')
```

This could also be written using argument substitution as:

```
scale_factor = 7.8
var = mds$value('\IP*$', scale_factor)
```

This feature is more commonly used with MDS\$PUT. To write an array called 'result' to a node called sig\_1:

```
mds$put, 'sig_1', '$', result
```

To associate a time vector with result a build signal expression would be employed:

```
mds$put, 'sig_1', 'build_signal($,*)', result, time
```

To read back the time vector, a function is used:

```
time = mds$value('dim_of(sig_1)')
```

Expressions which call into external functions provide an easy mechanism for accessing native mathematics or statistics libraries like IMSL<sup>5</sup>. The expression syntax is:

```
image->routine([arguments,...])
```

This has also been used to provide a bridge to legacy data. A tree can be constructed which organizes the available data items, and whose nodes contain external function calls which retrieve the actual data. Once this is done, all of the advantages of the data organization, and the simple homogeneous interface can be utilized. New analysis results can be written back into the trees along with the original data references.

#### 5 Relational Databases

Although they are not strictly part of MDSplus, relational databases play an important role in our overall data system. They are used by the data system for both administrative functions and to store data related to the

experiment. At Alcator C-Mod, one database contains the index of the archived pulses in the optical jukebox.<sup>6</sup> Another contains the electronic logbook, and various tables of numeric data extracted from the pulse files. Some of these are filled in interactively, others automatically between shots or at the end of the run day.

An interface, analogous to the one provided for MDSplus data, is provided for access to data in relational databases. The user provides an SQL statement with optional place holders for variable substitutions. If the statement is a SELECT then the selected columns are returned in trailing arguments to the routine. The following example selects plasma current and pulse length from the summary table for shots which meet some criteria.

```
Count = DSQL("select shot, max_ip, pulse_length from
summary where shot > 970600000 and max_ip > 1E6",
shots, ips, lengths)
```

In a similar fashion to MDS\$VALUE place holders could have been used in the query and program variables substituted for them.

To store data in the database an INSERT or UPDATE statement is used with the same routine.

```
Count = DSQL("insert into summary (shot, max_ip,
pulse_length) values (?, ?, ?)", shot_number,
plasma_current, pulse_length)
```

The logbook and its associated numerical tables has been implemented using this one interface. Figure 3 shows a screen from the new interface to the logbook.

## 6. Future directions

MDSplus has been operational for over eight years. As a mature system, the amount of effort required for routine maintenance is decreasing. Changes to the computing and research environment are driving the current projects. These include ports to new operating systems and relational databases, tools for remote collaborators and a new interface to the electronic logbook.

### 6.1 Port to New Operating Systems

MDSplus was originally implemented as a set of OpenVMS<sup>7</sup> applications. Developments in the computer industry and the interests of potential new users of the system have motivated a port to UNIX and WIN32<sup>8</sup>. The porting work is proceeding, starting with the pieces

that are needed by remote collaborators with non OpenVMS hosts, and will eventually include the entire package.

The API to the data is now available under UNIX and WIN32 using a client/server architecture.<sup>9</sup> Expressions are passed over the network to a server process which evaluates them and writes the result back. Using this the DWSCOPE has been ported to UNIX/MOTIF. This API has also been employed in personal computer based diagnostics. The complete expression evaluator is now working under UNIX and WIN32. The current plan is to attach this to a lower level client/server interface to the data.

### 6.2 WWW/JAVA Applications

A set of JAVA classes has been developed to provide applets with access to MDSplus data and events. These classes have been used to implement a WEB based SCOPE, experiment state display and live video from the control room and of the plasma. These tools allow remote collaborators to stay in close touch with the day to day operations of the experiment. They also serve an important public relations function. See: <http://www.pfc.mit.edu/cmod/> for links to these applets.

### 6.3 New Logbook Interface

C-Mod has been using an electronic logbook based on a relational database since it began operation. Two interfaces were provided, a Xwindows--based one written in IDL and a text--based one implemented on top of the VMS native editor TPU. Since we were unsure how the information would actually be used, the original tools were designed to be totally general purpose. While the tools were completely functional, they turned out to be awkward to use. This led to reduced user acceptance and utilization of the logbook. A new interface has been developed to address these issues. It is specifically tailored to making text based entries associated with an ongoing experiment.

The main screen of ENTRY\_DISPLAY presents the user with a dynamically updating set of entries which are of interest. As new records are added to the logbook, they are automatically added to the display. They can be selected, modified, and voided from this main screen. Figure 3 shows a typical screen. The selection criteria are flexible, but tailored to following along with the entries from an ongoing run. A typical selection might

be the users own entries and the session leader's from the current run which have not been voided. Both performance considerations for the automatic updates, and ease of use considerations, dictated set of queries supported.

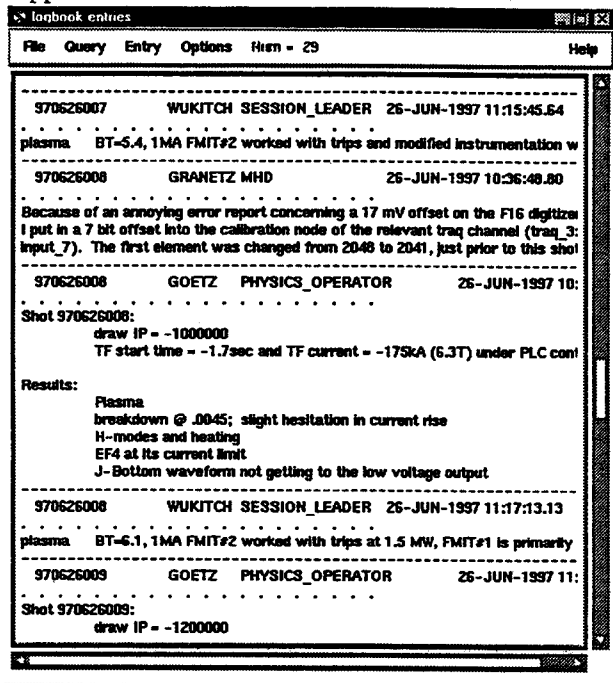


Figure 3 ENTRY\_DISPLAY logbook screen

This new interface has caused a dramatic increase in the use of the logbook during the current run period. There are plans to extend its functionality to allow for general queries with the automatic update feature disabled.

## 6.4 Database Migration

Relational databases are a critical part of the system. MDSplus was originally implemented using the RDB database from Digital Equipment Corporation. Personal computers and personal computer databases are becoming more and more powerful. RDB's dependence on OpenVMS and its relatively high cost, now that it has been sold to a third party, have motivated a search for a new solution. We plan to implement our current database interface on top of one of the inexpensive, PC-based databases.

## 7. Conclusion

MDSplus has proven to be an effective tool for data acquisition on a variety of experiments. The centralized data store allows large data sets to be structured in an

understandable way. This greatly lengthens the time that the data remain useful. The rich set of data types supported, allows for easy representation of all of the data from an experiment, from raw digitized signals to highly processed results. The expression based interface provides a simple mechanism for users and applications to access any of these data.

The current projects are an exciting next step in the development of the package. Moving to platform independence should both widen the appeal, and prolong the life of the package. Remote collaborations and remote operation of the experiment are sparking the development of new WWW interfaces and applications. These could form the basis of a fusion community wide data access system.

- 1 J.A.Stillerman, T.W.Fredian, K.A.Klare, G.Manduchi, "MDSplus Data Acquisition System", Review of Scientific Instruments, 68(1), January 1997, p939.
- 2 J.M.Schachter, "Local Transport Analysis for the Alcator C-Mod Tokamak", PFC/RR-97-2, pp217-265. MIT Plasma Science and Fusion Center, Cambridge, MA, June 1997.
- 3 S. HORNE, et al, "Performance of the C-Mod Shape Control System", Proceedings of the IEEE NPSS 14th Symposium on Fusion Engineering, 242 (1991)
- 4 IDL is a registered trademark of Research Systems Incorporated.
- 5 IMSL is a scientific subroutine library from Visual Numerics Inc.
- 6 T.W.Fredian, J.A.Stillerman, M.Greenwald, "Data Acquisition for Alcator C-Mod", Review of Scientific Instruments, 68(1), January 1997, p935.
- 7 OpenVMS is a registered trademark of Digital Equipment Corporation.
- 8 WIN32 - is a registered trademark of Microsoft Corporation.
- 9 T.W.Fredian, J.A.Stillerman, "Mdsplus Remote Collaboration Support - Internet and World Wide Web", these proceedings.