MDSplus

What is it and how did it happen?

Tom Fredian, Josh Stillerman, Gabriele Manduchi

Introduction

- MDSplus is a data handling system used widely in the fusion energy research community.
- I will attempt to describe many of the features of MDSplus which led to its popularity followed by some of its history and the approaches used in designing, developing and supporting the package.

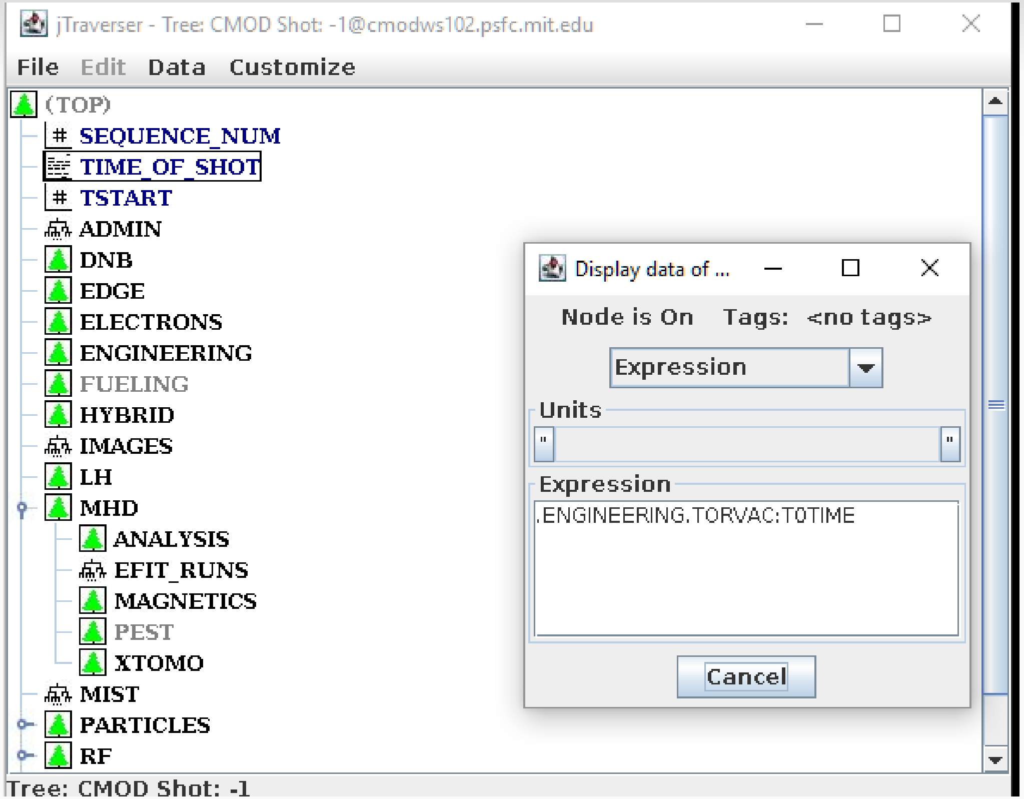
What is MDSplus?

- A collection of tools for data acquisition, analysis and visualization
- Data stored in hierarchical tree structures
- Rich data types
- Built-in expression evaluation
- Simple API's to store and retrieve data.
- Remote data access
- Open Source, Free to use
- More info at http://www.mdsplus.org

MDSplus Trees

- Hierarchical Data storage.
- Can contain a complete dataset of an experiment including setup information, actions to perform during and after the experiment, recorded measurements, analysis results and documentation.
- Model/Pulse file concept for pulsed experiments
  - Fully populated hierarchy in the “Model” tree
  - “Model” tree copied to the “Pulse/Shot” tree before the experiment cycle
  - Data from pulse stored into the “Pulse/Shot” tree.
- Concurrent reading and writing across multiple processes.
- Graphical and command based tools as well as language API’s for viewing or modifying the tree structure.
- Subtree concept.

MDSplus Trees - Traverser



jTraverser - Tree: CMOD Shot: -1@cmodws102.psfc.mit.edu

File Edit Data Customize

```
(TOP)
  SEQUENCE_NUM
  TIME_OF_SHOT
  TSTART
  ADMIN
  DNB
  EDGE
  ELECTRONS
  ENGINEERING
  FUELING
  HYBRID
  IMAGES
  LH
  MHD
    ANALYSIS
    EFIT_RUNS
    MAGNETICS
    PEST
    XTOMO
  MIST
  PARTICLES
  RF
```

Display data of ...

Node is On Tags: <no tags>

Expression

Units: ""

Expression: .ENGINEERING.TORVAC:T0TIME

Cancel

Tree: CMOD Shot: -1

C-Mod Experiment tree containing nearly 500K nodes.

Rich data types

- Scalars - Int8, Uint8, Int16, Uint16 … Uint64, Float32, Float64, Complex32, Complex64, Text
- Arrays of Scalars
- Signal - conversion expression, raw data, dimension info
- Range - start, end, inc
- Function - reference to internal or external functions
- Method - reference to DAQ device method
- Action - Job to be performed by MDSplus action server
- Node references
- List, Dictionary, and more

Expression Evaluator - “tdi”

- Loosely typed expression evaluation with large number of builtin functions
- User defined functions written in either tdi language or python
- Python statements can also be executed using a built-in py() function.
- Ability to call into C code in shared libraries.
- Expressions can be stored in tree nodes to be evaluated when a node is referenced.

Tdi expression use

- Used in API’s for data access

ans=MdsValue(‘node1 * node2/node3 - 42.’)

- Stored in tree nodes

If mynode contains an expression such as: sind(node1) * 100.
ans=MdsValue(‘mynode’) would return the result of the expression

- Signals (i.e. conversion from digitizer counts to volts)

API’s provide for many languages

- Simple API’s for several languages
  - C, Fortran, Java, IDL, Matlab, LabView
  - MdsOpen(‘tree’,shot), ans=MdsValue(expression[,args]) MdsPut(node,expression[,args]),MdsClose(),MdsConnect(connect-spec)
- Object Oriented API’s for other languages
  - Python, C++, Java, Matlab
  - t=Tree(‘tree’,’shot’), node=t.getNode(‘name’),contents=node.getRecord()

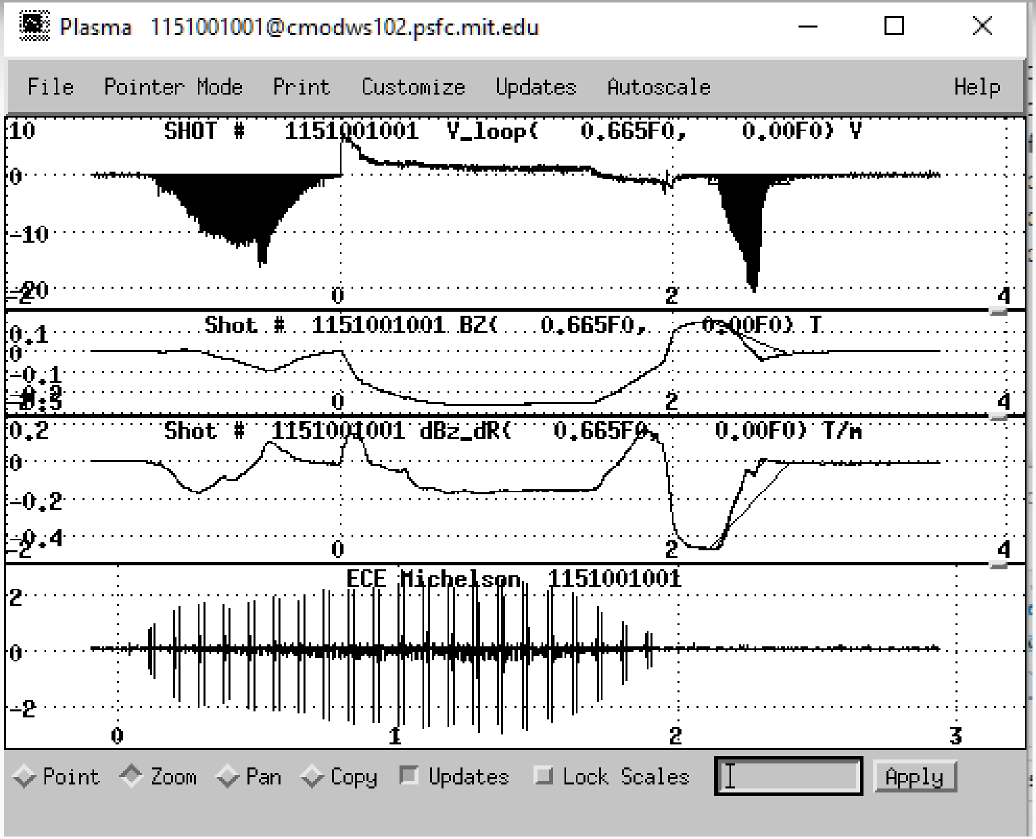
Remote Data Access

- “Thin” Client/server connections using plugin transport protocols
    - MdsConnect(‘host[:port]’)
    - MdsConnect(‘ssh://[user@]host’)
- “Distributed tree” connections using protocol plugins
    - Environment variable specifying tree file location
    - tree_path=’[connection-str]/dirspec[;[connection-str]/dirspec…]]’
    - cmod_path=alcdata-new::/cmod/trees/new/~t/;alcdata-models::/cmod/trees/models/~t/;alcdata-archives::/cmod/trees/archives/~i~h/~g~f/~e~d/~t

DAQ Devices

- Devices as collections of tree nodes with code to perform methods on the hardware (i.e. init, trigger, store,...)
- Device specific code written in any of C, tdi, Java or Python
- User sites can add site specific device support without any changes to MDSplus.
- Action descriptions in tree nodes used to dispatch the execution of device methods on servers connected to those devices.

Visualization tools


Plasma 1151001001@cmodws102.psfc.mit.edu
File Pointer Mode Print Customize Updates Autoscale Help
SHOT # 1151001001 V_loop(0.665F0, 0.00F0) V
Shot # 1151001001 BZ(0.665F0, 0.00F0) T
Shot # 1151001001 dBz_dR(0.665F0, 0.00F0) T/m
ECE Michelson 1151001001
Point Zoom Pan Copy Updates Lock Scales Apply

Dwsope (x11 app) and jScope (java)
Pan, Zoom,Auto update...

MDSplus API’s

- Many languages supported
  - C, C++
  - Fortran
  - Java
  - Python
  - Matlab
  - LabView
  - IDL
  - Julia
  - Remote Objects

Supported Platforms

- Software repositories providing MDSplus installers for numerous platforms
  - Ubuntu versions 12, 14, and 16 (32-bit and 64-bit)
  - Redhat Enterprise Linux versions 5, 6 and 7 (32-bit and 64-bit)
  - Fedora Core versions 17, 18, 20, 21, 22, 23, 25, 27 (32-bit and 64-bit)
  - Debian version Wheezy (32-bit and 64-bit)
  - RasberryPi debian (armhf)
  - Alpine Linux version 3.3 (32-bit, 64-bit and armhf)
  - MacOSX
  - Windows

Automated Build System

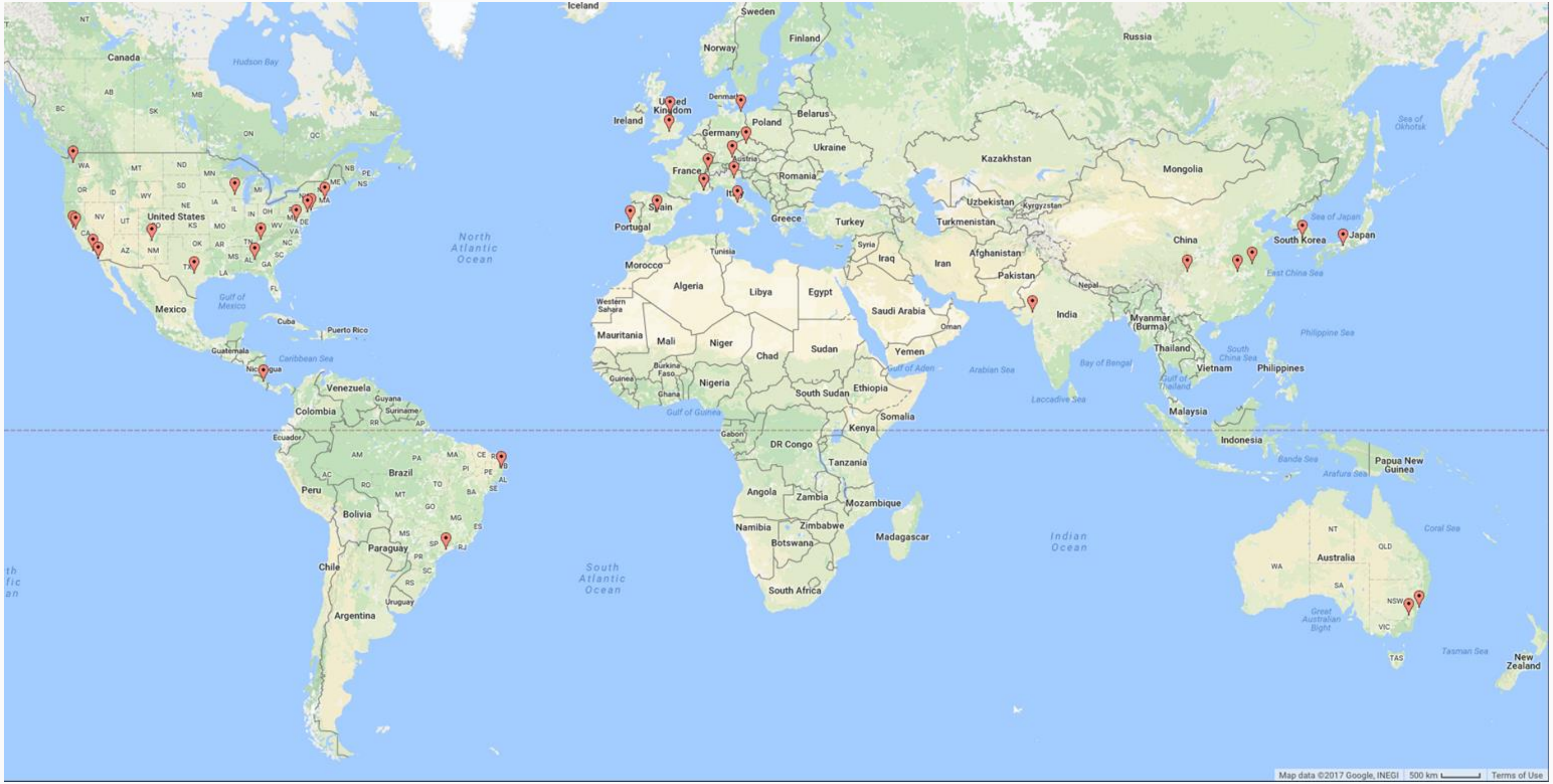
- Nightly release builds of most of the supported platforms
  - Code is tested on most platforms before new release is created.
  - Package contents verified before publishing a release.
- Automated test builds of GIT pull requests
  - Increasing set of regression tests performed on most platforms
  - Testing with tools such as valgrind and gcc sanitize to detect memory leaks, race conditions, uninitialized memory access etc.
  - Tests fail on compiler warnings
  - Changes not merged into main MDSplus branches unless all tests succeed.

Note: Use of GIT makes it easy for other users to suggest changes or report issues. Fix suggestions and bug reports are welcomed! (1145 Pull requests)

MDSplus Users

- MDSplus is now used at nearly 40 different research facilities worldwide.
- Many sites are now using MDSplus for data storage.
- Many sites that are using their own data storage software are using MDSplus to provide Internet access to their data.
- Scientists can use a common set of tools to access experiment and simulation data no matter where the data resides.
- MDSplus is also used for research unrelated to fusion such as space exploration. Based on analytics on downloads, MDSplus is used in many places which we know little about!

MDSplus Worldwide



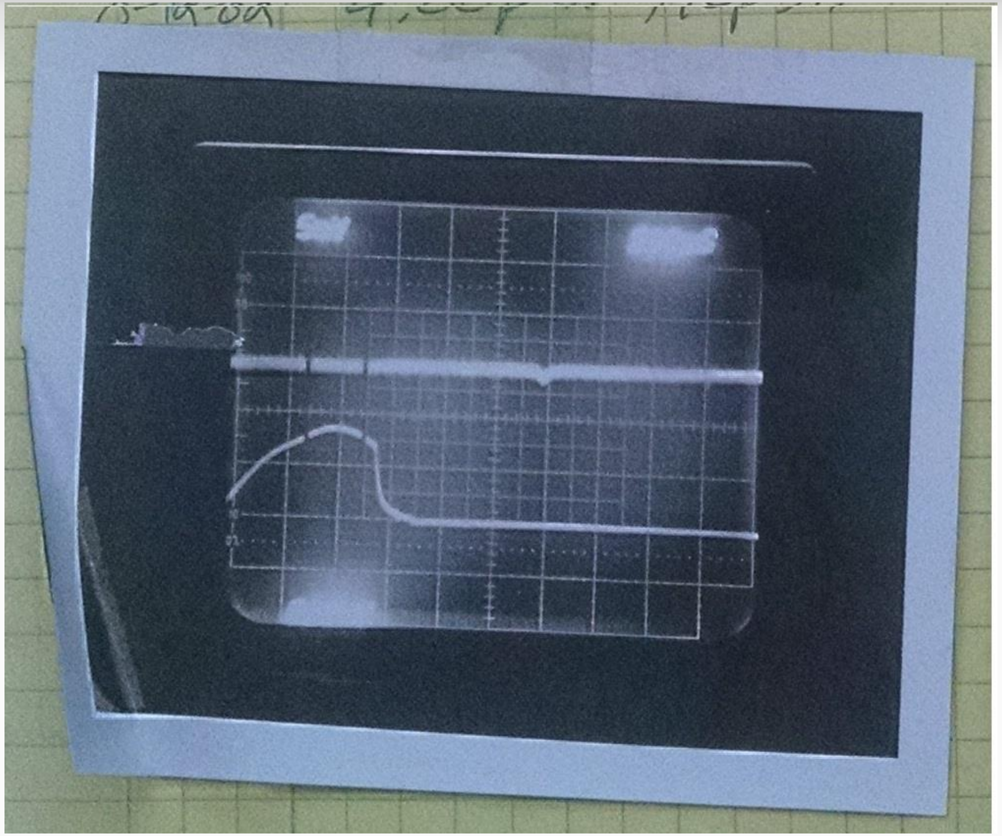
A bit of history...

- I enjoy reminiscing as I near retirement so please bear with me a little longer.
- Hopefully some of this will be amusing/interesting to you.

In the beginning… MDS

- I joined the Alcator experiment team at MIT in 1982.
- Josh Stillerman joined the Tara experiment team at MIT in 1982.
- Both hired to provide computer/programming help to the scientist and engineers at the two separate fusion experiments.
- My first impressions of Alcator experiment operations
    - Data shared among scientists by exchanging lists of numbers on paper.
    - Data acquisition: magnetic rotating drum recording 128 signals and 4-5 CAMAC modules.
    - Some data archived as Polaroid camera images of oscilloscope displays!

Lab notebook with photo



In the beginning… MDS

- I began a project to build a “data system” which would enable scientists to share data and reference measurements by name instead of a number from 1 to 128. I used experience with process control systems that I helped develop at my previous job with the DuPont company.
- Josh joined me on this effort (much to the dismay of some of the scientists on the Tara experiment because obviously the same data acquisition system could not be applicable to both mirror and tokamak experiments!).
- This system was later named MDS by members of the Tara experiment staff.

MDS - MIT/Model Data System

- Central storage of data accessed by named signals
- Simple expressions enabling conversion from digitizer counts to engineering/physics units (i.e. volts, temperature)
- Lossless data compression
- Data acquisition device support
- Written entirely in Fortran on VAX/VMS platform.
- Usage spread to many other fusion research facilities by 1987 including (USA) PPPL, ORNL, UCLA, SNL, U of Washington, Columbia University, U of Wisconsin, (Australia) ANU, (Japan) NIFS, (Sweden) KTH, (South Korea) NFRI.

MDS->MDSplus

- In 1987 three new experiments were being constructed, Alcator C-Mod, RFX and ZTH
- Padova expressed interest in using MDS but also had some DAQ ideas of their own. ZTH planned to use whatever RFX used.
- Discussions with RFX and ZTH lead to several new feature ideas for MDS
  - Tree structure
  - Greatly expanded expression capabilities
  - Much richer set of data types
  - All data including device setup, action dispatching info, calibration data, analysis results to be stored along with the data recorded from the data acquisition devices.
- Development began ~1988
  - Core development team: Tom, Josh (MIT), Ken Klare (LANL), Giulio Flor, Gabriele Manduchi(RFX).

MDSplus Used on C-Mod and RFX startup

- Three years after initial design discussions MDSplus was first used in 1991
- Successful startup of the C-Mod and RFX experiments acquiring data using MDSplus
- Those initial MDSplus data files are still accessible using the latest MDSplus releases.

MDSip - Internet Access to MDSplus

- Circa 1995 Remote Data Access implemented.
- A collaborator from University of Maryland requested a mechanism to access C-Mod data from his Linux system.
- An MDSip remote access library was then developed (TCP/IP only).
- This feature, quickly assembled for one user, would eventually become one of the most valuable features of MDSplus.

MDSplus/VMS -> MDSplus/Unix

- Circa 1997
- Team: Assembled with participants from MIT, PPPL, LLNL, ORNL and GA
- Goal: To make a version of MDSplus for Unix based platforms while retaining compatibility with VMS MDSplus
- Assumption: VMS will remain the dominant platform for running MDSplus
  - Single code base for use on both VMS or Linux
  - Emulate VMS system provided libraries and utilities on Linux

Strategy/Techniques used in Development

- Shared ideas and casual discussions
- Trial and error
    - Used “RAD” techniques (Rapid Application Development).
        - Decide on a feature.
        - Build a prototype.
        - Let people use it.
        - Fix/improve and repeat.
- No formal requirement or design documents
- Small core group of developers working well together
- Lots of useful feedback from the user community.

Stategy/Techniques (continued)

- Approach seemed to fit the fusion research environment
  - Needs of scientists and engineers continually (rapidly) evolving
  - Separation of DAQ from machine control
    - MDSplus originally designed for data acquisition and data analysis only
    - Machine control performed mostly by PLC’s or PCS applications which could access MDSplus data.
    - Flexibility to evolve MDSplus code without posing a danger to experiment or staff
  - New software and hardware technologies continually becoming available which could be quickly incorporated into MDSplus functionality (CPCI, Cameras, faster CPU’s and storage, Python, Java, Matlab, Labview…)
  - Ever growing customer base each bringing new ideas to affect the evolution of MDSplus

MDSplus Tomorrow

- Continued commitments from funding agencies for ongoing development and support of MDSplus
- New team members:
  - New ideas, knowledge of modern techniques for software development and support
  - Renewed enthusiasm for maintaining and improving MDSplus
  - Insurance for project longevity
- Ongoing work to improve the quality and robustness of MDSplus
- Ongoing plans to add new features as the needs of the fusion research community continues to evolve.

Conclusions

- RAD approach to developing MDSplus worked well
- MDSplus has become a valuable tool to the fusion research community
- Successful ongoing collaboration of several laboratories to build and maintain software for the entire community
- Built on feedback from the user community expanding the feature set to meet their needs
- Incredibly enjoyable experience being part of a great team that continues to work well together including several new participants to carry MDSplus on into the future!