Available online at www.sciencedirect.com

ScienceDirect

Fusion Engineering and Design 83 (2008) 317–320

Fusion Engineering and Design

www.elsevier.com/locate/fusengdes

MDSplus extensions for long pulse experiments

T. Fredian[a,*], J. Stillerman[a], G. Manduchi[b]

[a] *Massachusetts Institute of Technology, 175 Albany Street, Cambridge, MA 02139, USA*
[b] *Consorzio RFX, Euratom-ENEA Association, Corso Stati Uniti 4, Padova 35127, Italy*

Available online 24 October 2007

Abstract

MDSplus is a data acquisition and analysis software package used widely throughout the international fusion research community. It was originally designed for use on pulsed experiments where experiment measurements are typically acquired by external measurement devices with local memory and then gathered by the data system after the pulse has completed. Today, more and more fusion research devices are being constructed which have very long pulse lengths. It is no longer acceptable to wait until the pulse completes before acquiring the measurements and making them available to the researchers. To explore the possibility of adapting MDSplus for use on experiments with long pulses, we have designed and implemented some prototype extensions to make MDSplus more suitable for those types of experiments. These extensions may be applicable in other areas such as data handling for fusion modeling codes. To implement these extensions, an additional API was developed to enable applications to store data incrementally in a node. The data is stored in an indexed linked list of data segments in the MDSplus data file enabling efficient retrieval of subsets of the data within a specified time interval.
© 2007 Elsevier B.V. All rights reserved.

Keywords: MDSplus; Data acquisition; Data analysis; Long pulse

1. Introduction

MDSplus [1–3] is a data acquisition, data handling and analysis system used widely in the fusion community. It provides numerous powerful capabilities to simplify the process of storing measurements or analysis results in archives and to enable scientists to retrieve these data using a wide variety of programming languages and data manipulation tools. It provides a powerful expression evaluator which can be used to manipulate the data on retrieval and expressions can be stored in the archive to define new data items which are computed when retrieved. MDSplus also provides a remote data access capability enabling scientists to access data from anywhere on the internet. The MDSplus system runs on many computing platforms including Windows, MacOS, Linux and several other variants of UNIX. It is used at over 30 fusion experiments or modeling sites worldwide.

MDSplus was originally designed and implemented in the late 1980s by a team comprised of the people from the Massachusetts Institute of Technology (USA), Los Alamos National Laboratory (USA) and Istituto Gas Ionizzati (Italy). The system was designed to support three short pulse fusion experiments under construction, C-Mod, RFX and ZTH. At that time, fusion experiments typically experienced pulse lengths ranging from fractions of seconds up to a few seconds and generated less than 10MB of data per pulse. Fortunately the advances of computer hardware have kept pace with the growth of data handling demands of fusion devices over the last 20 years. Although fusion experiments now often generate many gigabytes of data per pulse, MDSplus still proves to be a powerful tool for handling data in fusion experiments.

2. New era of fusion experiments and modeling codes

Many fusion experiments recently built, under construction or planned such as LHD [4], W7X [5], EAST [6], KSTAR [7] and ITER [8] are designed to produce much longer duration pulses and in some cases approaching continuous operation. This greatly alters the data handling requirements compared to short pulse experiments. In short pulse experiments, the bulk of the data acquisition is done in a batch mode and occurs after pulse has completed. Since the pulses were short, scientists have

* Corresponding author. Tel.: +1 617 253 7623; fax: +1 617 253 0627.
E-mail address: twf@psfc.mit.edu (T. Fredian).

0920-3796/$ – see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.fusengdes.2007.08.022

been content to wait a minute or two before viewing or analyzing the data. With long pulse experiments, where the pulses may endure for many minutes or more, the quantity of data increases dramatically and the data must be acquired and accessed during the pulse.

A similar phenomenon is occurring in the modeling community. Codes are generating large quantities of data over long periods of time and much of the data must be monitored during the course of the code run.

Some sites, LHD [4] for example, have dealt with long pulses by approximating them as a series of adjacent short pulses. Other planned experiments such as W7X [5] and perhaps ITER are developing completely new data systems to handle quasi-continuous data flows.

3. The challenge

Since MDSplus is currently in use at numerous experiments worldwide and many of the scientists in the fusion community are familiar with its tools and data access capabilities, a significant investment in learning time of both the scientist and the support staff could be avoided if the MDSplus system could be enhanced to provide a mechanism for handling long pulse data streams while continuing to provide all of the powerful features found in the original MDSplus. In addition, utilizing MDSplus on these long pulse experiments would enable scientists worldwide to access the data using the same tools they currently use for accessing existing short pulse experiments.

The additional functionality needed for long pulse experiments or modeling codes should be added in such a way that minimizes the changes required in the MDSplus interfaces and retain compatibility with existing data archives. While other scientific storage methods such as HDF5 [9] provide very little backward compatibility with data sets constructed with earlier versions, MDSplus today can still access the original data files that were written dating back to the early 1990s. Any new enhancements for long pulse should not alter this policy of retaining compatibility with existing data or applications.

To provide most of the additional functionality required by long pulse experiments, MDSplus must be enhanced to be able to efficiently append data to a signal stored in a node of an MDSplus tree and to provide a mechanism for efficiently retrieving a subset of the data. While MDSplus currently provides the ability to retrieve sub-sampled data using the subscripting capability in the expression evaluator, MDSplus subscripting retrieves the entire signal from the data file and extracts the desired samples. The existing capability is therefore too inefficient for long pulse data.

The challenge is to provide this ability to append data and retrieve subsets of the data efficiently in MDSplus while maintaining compatibility with existing capabilities and archived data.

4. Segmented records

The main design concept arrived at for supporting quasi-continuous data in MDSplus is the use of “segmented records.” As data are acquired, the data are stored in segments in the MDSplus data file. A segmented record in MDSplus consists of an index of the data segments and the segments themselves. As new data arrives it is appended to the current data segment if space permits. If not, a new segment in the data file is allocated and added to the index and the new data is added to the new segment. The index of segments includes the start and end times of the segment represented as a 64-bit integer. The index provides an efficient mechanism for identifying and locating the segments of interest when retrieving the data. The size of each segment is specified in the API so the segment size can be optimized based on the data flow characteristics of each measurement. A data segment consists of rows of data where each row is associated with a single timestamp. A row in a segment can be a scalar value or a multidimensional array. Segment indexes are allocated in fixed size tables. When an index becomes full, an additional index is allocated in the data file with pointers back to the previous index.

Two different types of segment records have been implemented to handle two distinct data flow models expected to occur in long pulse experiments. On type of segmented record which was called “normal” segmented records for lack of a better name, is likely to be used when storing data from transient recorders which buffer up samples occurring at multiple timestamps. When storing data of this type, the application must provide a description of the timebase to be stored with the segment. The other segment record type, called “timestamped” segmented records, is used when storing measurements which more typically arrive one timestamp at a time. With this type of segment, when a new segment is allocated additional space is allocated to accommodate one timestamp per row. Data in a timestamped segmented record are added one row at a time.

Time is represented in the segmented records by 64-bit unsigned integers. The only stipulation with time values is that the times associated with rows of a segmented record must be increasing in value. MDSplus provides some built-in functions for obtaining 64-bit timestamps using the OpenVMS [10] time representation which is the number of 100 ns ticks since a base time (November 17, 1858). However, any timestamp representation can be used depending on the needs of the application. In some cases it may be convenient for the timestamps to represent the number of clock ticks since a base time such as 10 s before beginning of pulse.

Before retrieving data from segment records, the user can specify a “time context” which can include one or more of the following parameters; the start time, the end time and the delta time. Once a time context has been specified, future retrievals from segmented records will return a decimated signal created by selecting the segments lying within the start and end time boundaries, resampling these segments based on the delta time and then combining the reduced segments. MDSplus provides default functions for resampling the segments and assembling them into the resulting signal. However, these functions can be overridden on a per node basis by specifying the names of user provided expression evaluator functions as *extended node attributes* described below. This capability makes it possible to accommodate any site or node specific representation of time