

PSFC-JA-00-13

**WWW Interfaces for Runtime Relational Database
Applications**

J.A. Stillerman, T.W. Fredian, M. Greenwald

May 2000

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

This work was supported by the U.S. Department of Energy, Cooperative Grant No. DE-FC02-99ER54512. Reproduction, translation, publication, use and disposal, in whole or in part, by or for the United States government is permitted.

Submitted for publication to *Fusion Engineering and Design*.

WWW interfaces for runtime relational database applications*

Joshua Stillerman, Thomas W. Fredian, Martin Greenwald
MIT Plasma Science and Fusion Center Cambridge, MA 02139 USA

*Work supported under DOE. Cooperative agreement number DE-FC02-99ER54512

Abstract

The Alcator C-Mod group has been using an electronic lab notebook based on a relational database since the experiment started in 1991. The software, called logbook, provides a shared lab notebook abstraction for all of the participating scientists. It currently contains more than 37,000 text entries. This application and its underlying database schema are also being used at DIII-D and NSTX in the US, and installations at TCV at EPFL and H1 at ANU are in progress. A new WEB based user interface is being developed, modeled on the ENTRY_DISPLAY interface, which was written in IDL™. It provides complete read/write access to the entries table to any authorized user via a WEB browser. This interface places much lower demands on network and computing resources than the IDL/X-Windows based version.

ENTRY_DISPLAY is based around an output window that contains a running display of all of the entries which meet some user-specified criteria. As new entries are made, all running copies of the application automatically re-query the database and update their displays. This application was implemented using standard tools provided with the Microsoft SQLSERVER™ database, and Internet Information Server under Windows/NT. An application which accesses the 'Signals' table in this same database, implemented with the same technology, is publicly accessible at <http://red.psfc.mit.edu/logbook/signals.htm>. It provides an index to the names of the raw and processed data signals stored for the Alcator C-Mod experiment.

1 Background

A lab notebook is supposed to be a permanent archival record of notes and data, written at the time of the experiment. Access to these notes needs to be at least as good as, if not better than, the access to the raw and processed data from the experiment. With the increased number of offsite collaborators, good shared lab notebooks are becoming even more critical. The logbook at C-Mod has gone through a long evolution to get to its current functionality. This report details the most recent developments, primarily in response to the need for economical and effective remote access.

The remote access problem was solved by moving the interface to the web. The server side application is called WWW Entry Display. The client is very thin; the server dynamically generates web pages from the database contents. The user's browser simply needs to render the results.

Using a relational database to underpin the logbook gives it the necessary performance for this task. By modern standards, our databases are quite small. At C-Mod there is approximately one half of a gigabyte of data in our relational databases. These databases contain only highly processed data, user comments and data management information. All raw and processed time series data are stored elsewhere using MDSplus[1]. Modern relational databases, with their highly optimized indexing, provide very crisp response for databases of these sizes. For example, to retrieve the text from all entries in 1999 containing the string 'H-Mode' takes approximately 1 second. This query

returns 323 out of 5124 entries. Selecting all of the entries from a given day takes significantly less time.

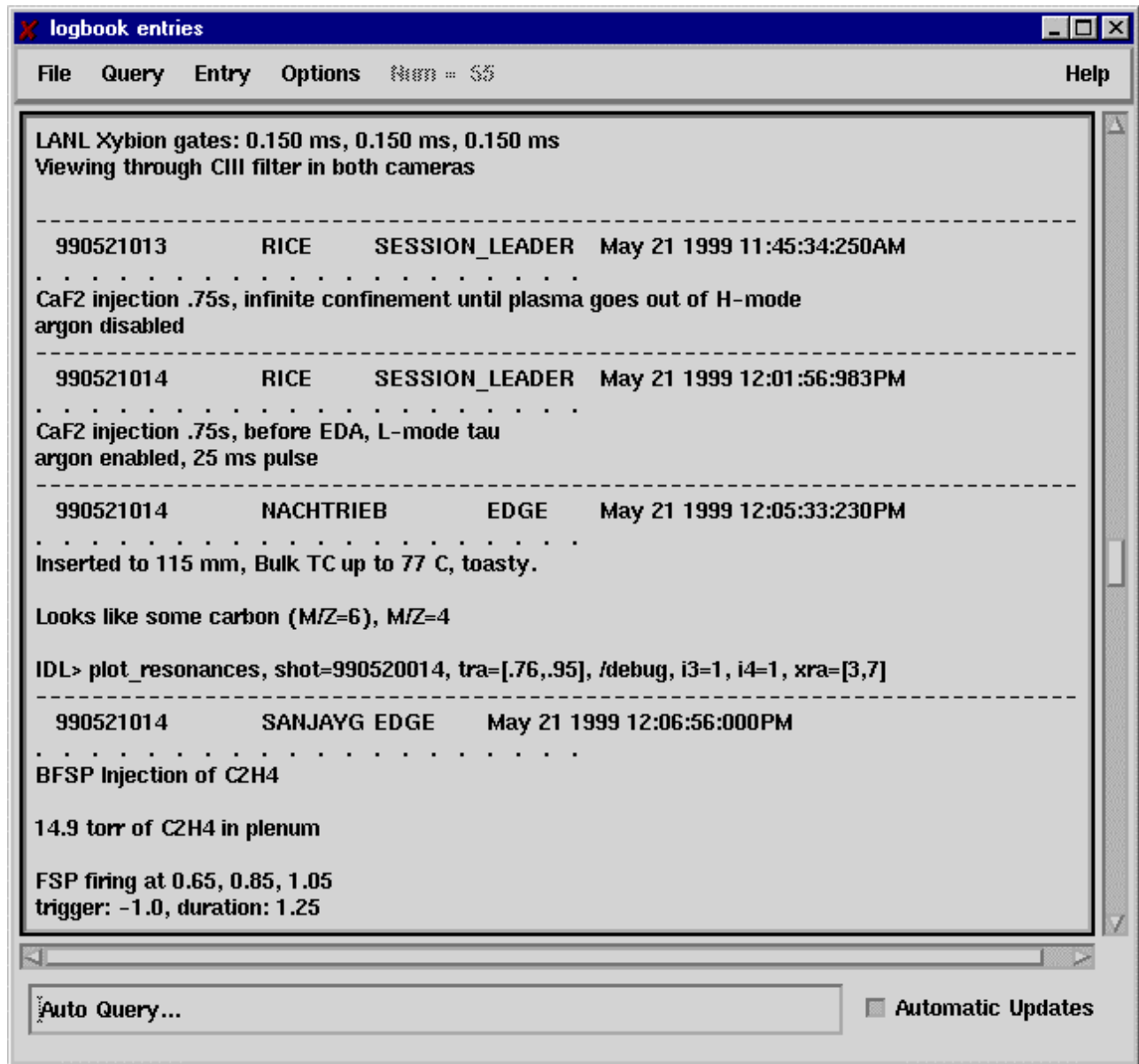


Figure 1 Original Entry Display

A good user interface is critical. If it is not intuitive and easy to use, it will not be used. The ENTRY_DISPLAY interface evolved with a lot of interaction with our user community. The logbook started with a text editor based user interface. This had the advantages of leveraging both the user community's familiarity with the editor, and all of the text manipulation and user interface code in the editor. The main difficulties with this implementation were that it was complicated to operate and entries were frequently made with incorrect qualifying shot, run, or topic. The next version was implemented using IDL widgets for X-Windows. This allowed us to provide a less cryptic but still somewhat awkward interface. This version erred on the side of too much flexibility, making it complicated to operate. The final design was arrived at after several years of

experience with these early interfaces. In particular, the users wanted to be able to see their comments as soon as they entered them. They also wanted to be able to correct them on the fly. Finally they wanted to be able to see a large number of entries at a time. ENTRY_DISPLAY, (see figure 1) written in IDL on top of the RDB relational database under OpenVMS was designed to meet these requirements.

With this new interface there was a dramatic increase in the use of the logbook. Usage went from an average of one or two entries per shot up to four or five per shot. The sale of the RDB database by Digital Equipment to Oracle raised some concerns about the long-term viability of the database. In 1998 the database was moved from RDB to SQLSERVER. The interface software was modified to use a client access library purchased from SYBASE. This new configuration was originally intended as only an insurance policy against the obsolescence of RDB. It was put into production because it turned out to have much better performance, be more robust, and included a large suite of tools for writing applications.

1.1 The Database

While the database engine was moved from OpenVMS/RDB to WinNT/SQLSERVER in 1998, the underlying schema for the logbook entries has not changed since its inception in 1991. The schema can be divided into two parts, the first is a fixed set of four tables that are used to manage user text entries. These are the tables that ENTRY_DISPLAY operates on. The rest of the database contains numeric tables that hold analyzed data, diagnostic configuration information, and shot summary data. These numeric tables are not accessed with this interface; instead they are accessed using the general IDL database interface [2], or any ODBC compliant database client. Using the same database for all of this information has the advantage that joins can be performed between these numeric tables and the entries table. For example, a user can find the entries that were entered since some run pertaining to shots with some particular plasma parameters and diagnostic configuration.

The four tables that are accessed directly by ENTRY_DISPLAY are:

1. Entries – holds the text (up to 8K bytes) qualified by username, topic, time entered, and typically run and shot.
2. Topics – contains a list of valid topic names for qualifying entries.
3. Runs – contains 1 record /run with the date, time, and run comment.
4. Shots – contains 1 record /shot with the date, time, and shot comment. Additional columns in this table are used to keep track of the data system performance, but they are not accessed by the interface.

A Pentium-based server, running SQLSERVER, with full text search enabled and indices on all of the relevant columns, provides very high performance. Typical queries complete in less than 200 msec. Network access to the database is implemented using SYBASE Open/Client library™, which is available for most platforms, or Microsoft's native dlibrary for PC platforms.

1.2 The Interface

ENTRY_DISPLAY is based around a main screen, which shows a running list of text logbook entries that are returned by a user specified query. As new records are added to the database, all

running copies of ENTRY_DISPLAY automatically re-query the database and update their displays. A dialog box helps the user specify their selection criteria. The predicates generated by this dialog are designed to operate well with the database's index structures. This makes the automatic re-querying for many users at a time reasonably inexpensive to do. A clause limiting the maximum number of rows returned is automatically added before the query is executed. Dialogs allow the user to specify the output format, and record ordering. A separate dialog allows users to enter other SQL selection predicates and optionally combine them with clauses from the selection dialog. Automatic updates are disabled when these 'custom queries' are in use, since they could potentially be expensive to execute. Figure 1 shows a typical Entry Display screen capture.

A user typically displays all the entries from the current run that he has entered merged with all the entries for selected topics. Records selected with the mouse can be either voided or modified by their original authors. No records are ever deleted from the database. They are just crossed out, and optionally re-entered with new contents. This is in keeping with the 'lab notebook' approach.

This application was written in IDL, and operates under OpenVMS, Tru64 UNIX, HP/UX, and WIN/32. It is actively being used at C-Mod, DIIIID and NSTX in the US. Installations are underway for TCV at EPFL and H1 at ANU. The design for the new WWW Entry Display comes directly from this original version.

2 WWW Entry Display

Offsite collaborators pose additional challenges to applications like the logbook. A 'lighter weight' version of ENTRY_DISPLAY was needed. The demands of either running X-Windows over a network, or worse yet telephone and modem, or having to have an IDL license and enough local CPU power to drive the X-Based application, seemed to be heavy a burden to put on our collaborators. A WWW based solution addresses these issues nicely. The WWW based version of ENTRY_DISPLAY is accessible from any web browser, making it platform independent. Since it is written as a server side application generating very standard HTML, it is also browser independent. The network load is significantly lower than the X-Windows based interface.

Security is a concern with such a widely accessible application. The built in security mechanisms in the web server address this nicely. It supports encrypted passwords, digital certificates, and even the use of SSL for all data transfers.

2.1 Interface

WWW Entry Display reproduces all of the functionality of the current IDL/X based interface on a remote web browser. The main screen of WWW Entry Display is made up of two frames, the top one contains user interface buttons and hidden refresher applet, which handles the automatic refresh. The bottom one contains a table with the results of the current query. Figure 2 shows a screen capture of this main screen. Pop-up browser windows handle other user inputs. User preferences are stored in a separate table in the database and are automatically loaded when the user first enters the page.

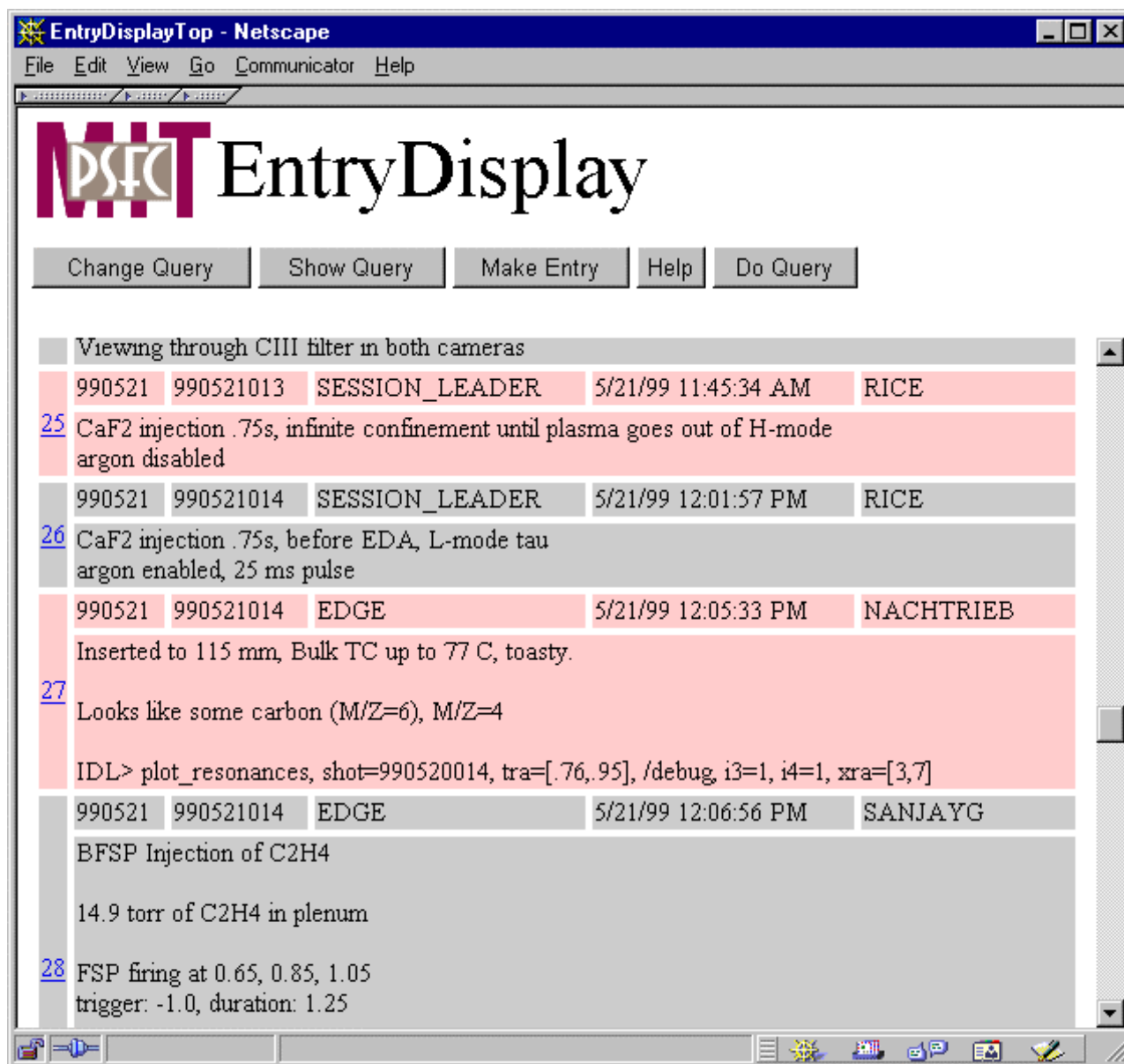


Figure 2 WWW Entry Display

2.2 Implementation

The implementation depends on only very standard web browser functionality. All of the computing, and all of the database specific code resides on the web server. The application is written in server side vbscript for Microsoft's IIServer. Running it on the server places the minimal burden on the client machine, and its browser's capabilities. The server side vbscript accesses the database using the built in ADO BD object set, and generates standard HTML tables with the results. Server session variables are used to pass information between the dialogs and to hold the state information for the application. User preferences are stored in an auxiliary table, indexed by username. When the user is authenticated, these saved preferences are automatically loaded.

Security is provided using the standard web server security mechanisms. If the browser supports it, Windows NT Challenge/Response Authentication is used. If not, then basic authentication is used.

NT automatically uses SSL protocol[3] for the dialogs it presents to users for changing their passwords. If more security is required, SSL could be used for all of the pages in the application. Finally digital certificates could be required to access the site. All of these mechanisms are provided with the web server. No additional code is required to use them.

An applet in the top frame connects to an MDSplus mdsip server[4] for event notification on changes to the database. When the event is received, the applet tells the lower, target frame to reload. It reconstructs the query out of the server variables, executes it, and outputs a new HTML table with the results. Each time a change is made to the database, an MDSplus event is generated. WWW Entry Display uses mdsip to distribute events when it modifies the database.

2.3 Status and Future Directions

A prototype of this application is up and running. It has not yet been released to the general user community. The screen layouts and the event notification system are still under development. The prototype has been used to provide access for scientists from their homes over low speed lines. The initial responses were very positive.

With the move to the WEB, it becomes fairly easy to incorporate additional types of information into the entries. In the future, entries could contain URLs, graphics, audio, and references to java scope applets [4]. The main difficulties with this will have more to do with the interface to create these items than with rendering them.

3 Other WWW database applications

This same technology can be used to provide access to other databases and tables. For example, at C-Mod we have a signals table that provides an index to all of the data items stored for the experiment. A web-based interface to this table is publicly available at <http://red.psfc.mit.edu/logbook/signals.htm>. Using this page, new users can locate signals of interest to them and even get a plot of sample data via a java scope applet.

Another application of this is our 'disk hogs' page which queries the archive database to display the top ten users of staging disk space for shot data. This page was put up to enable the community to better manage their disk resources. Applications like this are very easy to develop using these tools.

4 Conclusion

There were many reasons for moving ENTRY_DISPLAY from IDL to the World Wide Web. The web version requires very little software on the client in order to run. It is both platform, and even browser independent. This version is much less CPU intensive, and uses much less network bandwidth than its predecessor. It operates reasonably well over telephone and modem connections. WWW Entry Display is compatible with the IDL version and can be used concurrently. User displays will be updated when entries are made from either of the interfaces. Finally, the web provides interesting room for expansion. Entries could be augmented with pictures, sound, web references and plots of time-series data using the java scope applet.

The ENTRY_DISPLAY application has proven to be a very effective interface for the electronic logbook. It directly addresses the user's complaints about the previous implementations. It is heavily used at C-Mod and DIII-D, installed and planned for use by NSTX and being installed at TCV and H1. The port to the World Wide Web should make it even more appealing in the near future.

References

- [1] 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.
- [2] J.A. Stillerman, T.W. Fredian, The MDSplus data acquisition system, current status and future directions, Fusion Engineering and Design 43 (1999) 301-308.
- [3] A. Frier, P. Karlton, and P. Kocher, "The SSL 3.0 Protocol", Netscape Communications Corp., Nov 18, 1996.
- [4] T.W. Fredian, J.A. Stillerman, MDSplus remote collaboration support –internet and world wide web, Fusion Engineering