Logbook Reference Manual

Order Number: MDSplus—LOG001

January 28, 1993

This manual describes the Logbook Rdb database and its Text Processing Unit (TPU) interface. The Logbook is of interest to all wishing to examine shot data taken by the experimental plasma physics programs. These commands are in the SYS$LIBRARY:LOGBOOK.TPU command definition file.

Revision/Update Information: This is a new manual.
Operating System and Version: VMS Version 5.4 or higher
Database and Version: VMS/Rdb Version 4.0 or higher
Software Version: MDSplus Version 1.0 Field Test
Author: Kenneth A. Klare
Group P-4, Mail Stop E554
Los Alamos National Laboratory
January 28, 1993

The information in this document is subject to change without notice and should not be construed as a commitment by the Massachusetts Institute of Technology, the Los Alamos National Laboratory, or the Istituto Gas Ionizzati.

The software described in this document is supplied under a license and may be used or copied only in accord with the terms of that license.

No guarantee is made that the software is adequately or completely described in, or behaves in accordance with this document. The only responsibility for any error found in this document or the program is to try to correct the documentation or programming.

Acknowledgment: The prototype Logbook was from Josh Stillerman of MIT and helpful suggestions from Michele Bassan of RFX have been used.

Copyright ©1991, 1992 by Los Alamos National Laboratory

All Rights Reserved.

The following and others are trademarks of Digital Equipment Corporation:

<table>
<thead>
<tr>
<th>DEC</th>
<th>DECdecision</th>
<th>Rdb</th>
<th>VT</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEC/CMS</td>
<td>DECnet</td>
<td></td>
<td>VAX</td>
</tr>
<tr>
<td>DEC/MMS</td>
<td>DECwindows</td>
<td></td>
<td>VAX/VMS</td>
</tr>
</tbody>
</table>

Postscript is a registered trademark of Adobe Systems, Inc.
## Contents

<table>
<thead>
<tr>
<th>Preface</th>
<th>vii</th>
</tr>
</thead>
</table>

### INTRODUCTION

<table>
<thead>
<tr>
<th>1 GENERAL INTRODUCTION</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 NOTATION</td>
<td>4</td>
</tr>
<tr>
<td>3 GETTING SET UP</td>
<td>4</td>
</tr>
<tr>
<td>4 TABLES</td>
<td>4</td>
</tr>
<tr>
<td>5 SCREENS</td>
<td>5</td>
</tr>
<tr>
<td>6 MAKING ENTRIES</td>
<td>6</td>
</tr>
<tr>
<td>7 RANGES &amp; LISTS</td>
<td>6</td>
</tr>
</tbody>
</table>

### SUMMARY

<table>
<thead>
<tr>
<th>1 GENERAL SUMMARY</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 ENABLE SUMMARY</td>
<td>3</td>
</tr>
<tr>
<td>3 ENTRY SUMMARY</td>
<td>3</td>
</tr>
<tr>
<td>4 BROWSE SUMMARY</td>
<td>4</td>
</tr>
<tr>
<td>5 SQL SUMMARY</td>
<td>4</td>
</tr>
<tr>
<td>DESCRIPTIONS</td>
<td>LOG</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>CSQL</td>
<td>3</td>
</tr>
<tr>
<td>EDIT</td>
<td>5</td>
</tr>
<tr>
<td>HELP</td>
<td>6</td>
</tr>
<tr>
<td>INSERT ANSWER</td>
<td>7</td>
</tr>
<tr>
<td>INSERT QUERY</td>
<td>8</td>
</tr>
<tr>
<td>ISQL</td>
<td>9</td>
</tr>
<tr>
<td>LOGBOOK</td>
<td>11</td>
</tr>
<tr>
<td>LOOK</td>
<td>12</td>
</tr>
<tr>
<td>MAKE ENTRY</td>
<td>13</td>
</tr>
<tr>
<td>MAKE KEYWORD</td>
<td>14</td>
</tr>
<tr>
<td>MAKE RUN</td>
<td>15</td>
</tr>
<tr>
<td>MAKE SHOT</td>
<td>16</td>
</tr>
<tr>
<td>MAKE TOPIC</td>
<td>18</td>
</tr>
<tr>
<td>NEXT</td>
<td>19</td>
</tr>
<tr>
<td>NO</td>
<td>20</td>
</tr>
<tr>
<td>PRINT</td>
<td>21</td>
</tr>
<tr>
<td>SCRIPT</td>
<td>22</td>
</tr>
<tr>
<td>SELECT ENTERED</td>
<td>24</td>
</tr>
<tr>
<td>SELECT KEYWORD</td>
<td>25</td>
</tr>
<tr>
<td>SELECT ORDER</td>
<td>26</td>
</tr>
<tr>
<td>SELECT RUN</td>
<td>27</td>
</tr>
<tr>
<td>SELECT SHOT</td>
<td>28</td>
</tr>
<tr>
<td>SELECT TOPIC</td>
<td>29</td>
</tr>
<tr>
<td>SELECT USERNAME</td>
<td>30</td>
</tr>
<tr>
<td>SELECT VOIDED</td>
<td>31</td>
</tr>
<tr>
<td>SET RUN</td>
<td>32</td>
</tr>
<tr>
<td>SET SHOT</td>
<td>34</td>
</tr>
<tr>
<td>SET TOPIC</td>
<td>36</td>
</tr>
<tr>
<td>SHOW COLUMNS</td>
<td>37</td>
</tr>
<tr>
<td>SHOW ENTRIES</td>
<td>38</td>
</tr>
<tr>
<td>SHOW GROUPS</td>
<td>39</td>
</tr>
<tr>
<td>SHOW KEYWORDS</td>
<td>41</td>
</tr>
<tr>
<td>SHOW RUNS</td>
<td>42</td>
</tr>
<tr>
<td>SHOW SHOTS</td>
<td>43</td>
</tr>
<tr>
<td>SHOW TABLES</td>
<td>44</td>
</tr>
<tr>
<td>SHOW TOPICS</td>
<td>45</td>
</tr>
<tr>
<td>SHOW VIEW</td>
<td>46</td>
</tr>
<tr>
<td>UNVOID</td>
<td>47</td>
</tr>
<tr>
<td>VOID</td>
<td>48</td>
</tr>
<tr>
<td>ZERO</td>
<td>49</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>APPENDIX A REQUIREMENTS</th>
<th>A–1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1 QUOTAS</td>
<td>A–1</td>
</tr>
</tbody>
</table>

iv
Preface

The Logbook is an interface to an Rdb database of information about shots, runs, and general machine conditions found in the experiment. This TPU interface allows you to interact manually with the database. You can get information from or put information into the database with full TPU editing capabilities.

Logbook makes use of many VAX/VMS operating system routines. It uses the shared image SYS$SHARE:TPUSQL.EXE and the TPU command file SYS$LIBRARY:LOGBOOK.TPU. It receives support from MDSplus shared image SYS$SHARE:MDSSQLSHR.EXE. You will usually use the system TPU section file for the Extensible VAX Editor (EVE).

This preface talks about:

• Intended Audience
• Document Structure

Hint: For the Bookreader, you should enable Hot Spots in the View menu of the manual to see crossreferences.

Intended Audience

This manual is intended for all users of MDSplus. The use of Logbook is mostly independent of the other MDSplus operations.

Familiarity with the experimental machine terminology will be needed to make full use of searching and to understand what others have said.

Experience with a screen editor is needed and experience with EDT or EVE is particularly useful.

Document Structure

This document contains descriptions of the Logbook commands used in interacting with the Logbook database.

The following is a list of the parts of this manual.

• Part II, Summary—The categories of available commands.
• Part III, Commands—The detailed descriptions of the commands including argument descriptions and some examples of their use.
• Appendix A, Requirements—The system requirements.
• Appendix B, SQL Information—Brief listings of the SQL commands.
• Glossary—Brief explanations of the terms.
Introduction

This section introduces you to the Logbook. It has these sections:

• Section 1, General Introduction—gives some general words.
• Section 2, Notation—talks about how commands as written.
• Section 3, Getting Set Up—tells you how to get started.
• Section 4, Tables—says what is in the tables.
• Section 5, Screens—tells you what you find on the screen.
• Section 6, Making entries—talks about adding data.
• Section 7, Ranges & Lists—describes how to select.
1 General Introduction

The Logbook gives you many commands to help in accessing information stored in the on-line database. You only need to learn the general syntax to use much of the interface to Rdb found in the Logbook.

There is help available within the editor. This document is available as a help library, a Bookreader book (using DEChwiness or X-windows), and in Postscript printable form.

The format and arguments of the commands are described in detail in Part III, Commands section of this document.

The Logbook-TPU interface accesses data stored in a VAX relational database (Rdb) by issuing the Structured Query Language (SQL) commands from simpler settings and selections. You can give any dynamic SQL command but they can be formidable on first sight and forming these commands might inhibit you from trying to use the database. (We want you to use it.)

Note: To acquaint you with SQL, the query is usually shown on the message line or in a buffer.

The Logbook interface packages many of the required commands so you don't have to think about them and instead you can go directly to the task of understanding what was happening on that day. For example,

(Do) SELECT ENTERED YESTERDAY:TODAY
(Do) SHOW ENTRIES

The result is a display of all entries made from the beginning of (midnight) yesterday through midnight this morning, that is, what happened yesterday. Use YESTERDAY:TOMORROW to see all that happened since yesterday.

Some things have been done for you:

- The SQL command "Select * From ENTRIES Where Entered Is Between 'YESTERDAY' And 'TOMORROW' And Voided IS NULL Order By Topic, Run, Shot, Entered, Username" was given.
- The inquiry result was formatted by columns in names and values.
- The SHOW buffer was used to display the answer.

You can cut and paste from the SHOW buffer to enter this in some other document or (Do)PRINT a portion or all of the buffer.

Notes: Each operation is automatically followed by an SQL COMMIT if successful or ROLLBACK if it errs. Thus multiple operations cannot be committed together, but this should not be needed by a Logbook user. This method assures the database is locked for the least time.

The commands could have been (Do)select y:tod and (Do)show e and the same result would have been displayed. This abbreviation is a kindness from EVE.
2 Notation

(Do)XYZ means you should strike the key labeled "Do" or its substitute on other keyboards and then type the command "XYZ". The (Help) key displays titles for the function keys, whereas (Do)HELP displays how to get information on the commands.

Gold-X means you should strike the PF1 or Gold keypad key and then strike the single key X.

PF2 toggles on and off the message buffer where you can see your mistakes and any comments from the program.

PF1 to PF4 refer to the top set of keypad keys. The numbered keys on the keypad are KP0 to KP9. They are often used with the Gold prefix key to do EVE commands.

Notes: The Logbook windows set the EDT keypad.

If you have a VT-100 keypad you need to use PF4 for (Do) key before the EDT keypad is set, that is, to do the (Do)LOG command. After the EDT keypad is set use the Gold-KP7 key sequence for (Do). Better yet, get a full keyboard or define a key.

3 Getting Set Up

You can create a command file, say SYS$LOGIN:LOGBOOK.COM, to define the logical names and do a command. It needs:

$ DEFINE TPU$CALLUSER SYS$LIBRARY:TPUSQL.EXE/NOLOG
$ DEFINE SQL$DATABASE disk:[area.LOGBOOK]LOGBOOK/NOLOG
$ EDIT/TPU/COMMAND=SYS$LIBRARY:LOGBOOK 'P1'

and you must fill in the "disk" and "area" names.

Say $ @LOGBOOK to start up TPU. You can then do a (Do)LOGBOOK (this can be abbreviated as (Do)LOG or anything between) command in TPU to display the entry screens and define the special keys or just start in with a (Do)SET SHOT or (Do)SET TOPIC command.

Use (Do)LOOK if you are browsing and not saying anything. Here you could start with a (Do)SELECT SHOT or similar command.

4 Tables

The Logbook database has several tables:

- RUNS—the run numbers and a brief description.
- SHOTS—the shot numbers, their run, and a brief description.
- TOPICS—the topic names and a brief description.
- KEYWORDS—the keywords and a brief description.
- ENTRIES—the text, its topic, optional run and shot numbers, and voiding date.
Each table also has the name of the user and the date that the row was entered (added).

The columns in the tables are shown in Table 1, Columns in the Tables and the constraints are listed in Table 2, Constraints in the Tables.

Table 1  Columns in the Tables

<table>
<thead>
<tr>
<th>column</th>
<th>RUNS</th>
<th>SHOTS</th>
<th>TOPICS</th>
<th>ENTRIES</th>
<th>KEYWORDS</th>
<th>example</th>
</tr>
</thead>
<tbody>
<tr>
<td>USERNAME</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>KKLARE</td>
</tr>
<tr>
<td>ENTERED</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>17-JAN-1992.14:52</td>
</tr>
<tr>
<td>RUN</td>
<td>yes</td>
<td>yes</td>
<td>-</td>
<td>opt</td>
<td>-</td>
<td>920117</td>
</tr>
<tr>
<td>SHOT</td>
<td>yes</td>
<td>yes</td>
<td>-</td>
<td>opt</td>
<td>-</td>
<td>920117001</td>
</tr>
<tr>
<td>TOPIC</td>
<td>-</td>
<td>-</td>
<td>yes</td>
<td>yes</td>
<td>-</td>
<td>XRAY,x-ray</td>
</tr>
<tr>
<td>BRIEF</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>-</td>
<td>yes</td>
<td>guess</td>
</tr>
<tr>
<td>TEXT</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>yes</td>
<td>guess,found</td>
</tr>
<tr>
<td>KEYWORD</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>yes</td>
<td>reversal</td>
</tr>
<tr>
<td>VOIDED</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>yes</td>
<td>yester.tomorrow</td>
</tr>
</tbody>
</table>

Table 2  Constraints in the Tables

<table>
<thead>
<tr>
<th>column</th>
<th>constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUNS.RUN</td>
<td>unique</td>
</tr>
<tr>
<td>SHOTS.SHOT</td>
<td>unique</td>
</tr>
<tr>
<td>TOPICS.TOPIC</td>
<td>unique</td>
</tr>
<tr>
<td>KEYWORDS.KEYWORD</td>
<td>unique</td>
</tr>
<tr>
<td>SHOTS.RUN</td>
<td>in RUNS.RUN</td>
</tr>
<tr>
<td>ENTRIES.TOPIC</td>
<td>in TOPICS.TOPIC</td>
</tr>
</tbody>
</table>

if ENTRIES.SHOT is inserted

ENTRIES.RUN = SHOTS.RUN where SHOTS.SHOT = ENTRIES.SHOT

5

Screens

There are two screen layouts.

• For data entry, the layout has three screen areas for buffers:
  • LOGBOOK_INFO displays the maximum shot and the lists of required topics and suggested keywords that were defined for this experiment.
  • LOGBOOK_SETTINGS displays the currently set run, shot, and topic and the rest of the script.
  • LOGBOOK_xyz, where xyz is the topic that is set. This writable buffer holds the new text for the ENTRIES table of the database.

• For data looking (browsing), the layout has two read-only buffers:
  • LOGBOOK_INFO displays as given before.
• LOGBOOK_SELECTION displays the selection criteria that are set for the SHOW commands. The criteria are TOPIC, RUN, SHOT, ENTERED, USERNAME, VOIDED, and KEYWORD. The sort ORDER may also be selected. The appropriate selections are applied to the various tables that can be shown. All selections apply to the ENTRIES table.

These special words may be used with many of the column selections:
• ALL makes no selection on that column.
• MAXIMUM designates the maximum shot or run entered.
• NEXT designates the maximum shot or run entered plus one.
• NOT NULL requires an entry in that column.
• NULL requires that item in that column be empty.

6 Making entries

Before you can give (Do)MAKE ENTRY, you must have given (Do)SET TOPIC and if you are writing about a shot, you must have given (Do)SET SHOT. Similarly, if you are discussing a run, you must have given (Do)SET RUN. The run, shot, or topic must exist with one exception: you can write about the next (current maximum plus one) shot.

New RUN, SHOT, TOPIC, and KEYWORD rows can be added by privileged users, SYSTEM, or the machine operation as determined by privileges of the database. The number of TOPIC and KEYWORD entries will likely grow very slowly with time. The number of SHOT and RUN entries will grow linearly with time.

Note: Whereas the other (Do)MAKE commands ask for the value needed with an appropriate default like those for (Do)SET commands, the (Do)MAKE ENTRY command assumes the (Do)SET commands were given.

Hint: These three (Do)SET commands and (Do)MAKE ENTRY have two-keystroke abbreviations using the Gold key. The (Do)INSERT commands and (Do)SHOW VIEW, and (Do)ZERO ENTRY also have Gold key abbreviations.

7 Ranges & Lists

For each (Do)SELECT commands you can specify a comma-separated list of elements. Satisfying any element satisfies the selection (OR).

Each element of the list are shown in Table 3, Selection Elements.
Alphanumeric ranges are compared as if lengthened with blanks.

Dates have an implicit time of 00:00:00.00. Do not use colon as the separator between the date and time because it gives hour zero, use period or dash.

Interior tabs and multiple spaces are replaced by a single space.

KEYWORD searches cannot cross the ends of lines and are case insensitive.

Each (Do)SELECT command (except (Do)SELECT ORDER) gives the number of qualifying rows in the ENTRIES table without other selections. This can tell you if your selection is too tight or too loose. The (Do)SHOW ENTRIES commands will display entries corresponding to the intersection (AND) of all the selections and thus may be empty even if each (Do)SELECT reported a nonzero count.
Summary

The commands grouped by their function in this summary.

Hint: For the Bookreader, you should enable Hot Spots in the View menu of the manual to see crossreferences.
1 General Summary

This part logically groups the Logbook TPU Interface (Logbook) commands. You should get acquainted with these groups because it is easier to find the routine you will need if you can identify it with its general category.

The format and arguments of the commands are described in detail in the Part III, Commands section.

Note: Although commands given to TPU/EVE can be abbreviate, the names of selected tables, columns, and topics cannot be shortened. Keyword searches look for what you give.

These are the groupings of the commands:

- Section 2, Enable summary—to choose the first display.
- Section 3, Entry summary—to make new Logbook entries.
- Section 4, Browse summary—to search the Logbook for earlier entries.
- Section 5, SQL summary—to do Structure Query Language inquiries and some general database operations directly.

Hints: The commands below are Hot Spots in the Bookreader.

The EVE commands (Do)BUFFER SHOW will let you look at the SHOW buffer after it has been removed. PF2 or (Do)BUFFER MESSAGES will let you look at the MESSAGES buffer to review your errors.

2 Enable summary

These commands set up the screens or switch back to them.

LOGBOOK LOOK
NO HELP

The (Do)LOGBOOK command displays the data entry form. The (Do)LOOK command displays the selection form. The (Do)NO LOGBOOK command reverts to the display before the (Do)LOGBOOK or (Do)LOOK command. A (Do)LOGBOOK or (Do)LOOK command is required once to enable the (Do)HELP LOG command.

3 Entry summary

These commands help with adding to the Logbook.

SET RUN SET SHOT SET TOPIC
MAKE RUN MAKE SHOT MAKE TOPIC
MAKE ENTRY MAKE KEYWORD
ZERO SHOW VIEW
The (Do)SET commands enable information for the (Do)MAKE ENTRY command and select entries for the (Do)SHOW VIEW command. The other (Do)MAKE commands are restricted and add new entries with BRIEF descriptions to the tables.

The keystrokes for typical data entry with a (Return) on each line might be:

(Do)SET TOPIC xray
(Do)SET SHOT 1234
... type text ...
(Do)MAKE ENTRY

The Gold-key form could be:

Gold-T Xray
Gold-S 1234
... type text ...
Gold-E

4

Browse summary

These commands allow searching of the Logbook database constrained by the selections made.

SELECT RUN          SELECT SHOT          SELECT TOPIC
SELECT ENTERED      SELECT USERNAME      SELECT VOIDED
SELECT KEYWORD      SELECT ORDER
SHOW ENTRIES       SHOW KEYWORDS
SHOW RUNS          SHOW SHOTS          SHOW TOPICS
EDIT               UNVOID             VOID

The (Do)SELECT commands enable qualifiers to the other commands. The most important command is (Do)SHOW ENTRIES because this is the command that shows the text.

The keystrokes of a browsing with returns on each line might be:

(Do)SELECT TOPIC xray
(Do)SEL SHO 1234
(Do)SHOW ENTRIES

followed by cursor and page moves to look at the full display.

5

SQL summary

These commands allow general, database, and SQL operations. A partial list of SQL commands and expressions and more is found in Appendix B, SQL Information.

CSQL              ISQL
INSERT ANSWER     INSERT QUERY
SHOW COLUMNS      SHOW GROUPS      SHOW TABLES
PRINT             SCRIPT           NEXT
The (Do)CSQL and (Do)ISQL commands display (in the SHOW buffer) the results of a general inquiry. The resultant query and its answer can be put in any text buffer with the (Do)INSERT commands.

These (Do)SHOW commands are special Rdb queries that display information about the database.

The (Do)PRINT command lets you print portions of any buffer.

The (Do)SCRIPT command allows you to write TPU programs with pauses for each line.
Descriptions

Here are the detailed descriptions of the commands that form the Logbook program.
CSQL

Does a general SQL command and displays the result of an SQL SELECT with column formatting.

**FORMAT**

```
(Do)CSQL  string
```

**ARGUMENTS**

`string`

Specifies the SQL command to be given to the interpreter.

**RELATED COMMANDS**

ISQL displays with a label in-line with each value, rather than above the column. (Do)ISQL is better for a wide columns or few row.

**DESCRIPTION**

The string is passed to the dynamic Structured Query Language interpreter. The query and its answer (if any) are presented to you in the system buffer named SHOW.

**Notes:**

- Only SQL SELECT commands return information but all CSQL commands display the TPU SHOW buffer.
- The buffer display returns to what you were looking at when you enter a (Do) command including the null command.
- The layout of the SHOW buffer will depend on the display width.

**Features:**

- COMMIT or ROLLBACK are automatically done for each command.
- An EVE-ism allows omitting quotes on the final argument of a command at the keyboard.

**Limitations:**

- Unlike in the $ MCR SQL$ program, the command must be spelled in full in dynamic SQL and no semicolon (;) is used.
- You cannot do SHOW, HELP, and SET operations because they are internal to $ MCR SQL$. 
- The command must be dynamically executable; these are not: CLOSE, DECLARE CURSOR, DECLARE STATEMENT, DECLARE TABLE, DESCRIBE, FETCH, FINISH (because it closes all cursors including its own), INCLUDE, OPEN, PREPARE, RELEASE, SELECT ... INTO (singleton select), or WHENEVER. Most of these are VAX extensions that are involved in the dynamic processing.
Logbook Commands
CSQL

EXAMPLE

(Do)Command: CSQL SELECT * FROM RUNS

This gives

SELECT * FROM RUNS
--------------------------------------------
Row USERNAME ENTERED RUN
BRIEF

A test of the MAKE RUN command.
EDIT

Voids (marks as bad) the selected entries, does the settings, inserts the query answer from the voiding, and allows you to edit until you give the command (Do)MAKE ENTRY.

FORMAT

(Do)EDIT [ENTRY]

prompts

Void all these entries? (Yes/No)[N]

RELATED COMMANDS

VOID and UNVOID are used to hide and unhide entries.

DESCRIPTION

You will usually do this command from the selection screens after doing (Do)SELECT to pick the faulty entry or entries.

This command:

• Does a (Do)VOID ENTRY to mark all selected entries as bad.
• Asks for and does (Do)SET TOPIC, (Do)SHOT, and (Do)RUN with defaults from the selections.
• Adds as in the (Do)INSERT ANSWER command to the display of the TOPIC voided.
• Allows you to edit the entry until you (Do)MAKE ENTRY.
• Lets you to use the PF4 key, Delete Line, to remove extraneous (sorry about that) text. You will be making other changes.

Note: All entries, voided or not, can be seen if you give the command (Do)SELECT VOIDED ALL. Because the default selection is VOIDED IS NULL, users do not usually see Logbook entries that have been voided.
HELP

Displays information on how to use the editor (TPU), the section file (EVE), or the Logbook.

These commands are in the SYS$LIBRARY:LOGBOOK.TPU command definition file.

FORMAT

(Do)HELP  [LOGBOOK] [topic] [subtopic] ...
(Do)HELP  [LC] [command]

prompts

Type a topic name or ? for a list:

ARGUMENTS

LOGBOOK

Specifies that Logbook help items are used. May be shortened to LO.

topic

Specifies the commands or informational subtopics of the Logbook help is to display.

RELATED COMMANDS

(Do)HELP TPU will get information on the TPU command set. (Do)HELP gets information on the EVE interface.

DESCRIPTION

Omitting all arguments gives the standard EVE help information on its commands and how to get to other (TPU and Logbook) help.

Omitting the topic gives the most general information about the Logbook.

The commands (Do)HELP LOGBOOK COMMANDS or (Do)H L C will give the list of Logbook commands.

You can get other information on MDSplus. Give (Do) HELP MDSplus for a list.

Note: You must give LOG in each help request for Logbook information or LC for Logbook commands.

EXAMPLES

1  (Do)Command: HELP LOG SQL

2  (Do)Command: HELP LC CSQ
INSERT ANSWER

Copies the SHOW buffer into the current editing buffer at the cursor.

FORMAT

(Do) INSERT ANSWER

RELATED COMMANDS

INSERT QUERY copies only the lastest inquiry at the cursor.

DESCRIPTION

Usually the SHOW buffer will have the result of the latest inquiry from a standard command or an explicit SQL query.

Note: Any SHOW command or changes to the SHOW buffer will also be copied.
INSERT QUERY

Copies the saved value of the latest SQL inquiry into the current editing buffer at the cursor.

FORMAT  (Do)INSERT QUERY

RELATED COMMANDS  INSERT ANSWER inserts the SHOW buffer, which can have the latest inquiry and the response.

DESCRIPTION  The inquiry could be from any of the commands that do database operations. The SHOW commands clearly do a database operation. More subtle is that the SELECT commands do one to get the count of rows satisfying the query.
ISQL

Does a general SQL command and displays the result of an SQL SELECT with column formatting.

**FORMAT**

```
(Do)ISQL  string
```

**ARGUMENTS**

*string*

Specifies the SQL command to be given to the interpreter.

prompts

```
Command:
```

**RELATED COMMANDS**

CSQL displays with the labels above the columns rather than in-line with each value. (Do)CSQL is better for a few narrow columns and many row.

**DESCRIPTION**

The "string" is passed to the dynamic Structured Query Language interpreter. The query and its answer (if any) are presented to you in the system buffer named SHOW.

Notes:

- Only SQL SELECT commands return information but all commands display the TPU SHOW buffer.
- The buffer display returns to what you where looking at when you enter a (Do) command including the null command.
- The layout of the SHOW buffer will depend on the width of the display.

Features:

- COMMIT or ROLLBACK are automatically done for each command.
- An EVE-ism allows omitting quotes on the final argument of a command at the keyboard.

Limitations:

- Unlike in $MCR SQL$, the command must be spelled in full.
- You cannot do SHOW and SET operations because they are internal to $MCR SQL$.
- The command must be dynamically executable; these are not: CLOSE, DECLARE CURSOR, DECLARE STATEMENT, DECLARE TABLE, DESCRIBE, FETCH, FINISH (because it closes all cursors including its command), INCLUDE, OPEN, PREPARE, RELEASE, SELECT ... INTO (singleton select), or WHENEVER. Most of these are VAX extensions involved in the dynamic processing.
EXAMPLE

(Do)Command: ISQL SELECT * FROM RUNS

This gives

SELECT * FROM RUNS
--------------------------------------------
------------------
RUN: 1
BRIEF: A test of the MAKE RUN command.
LOGBOOK

Initializes the Logbook variables, keys, and help and displays the data entry screens.

FORMAT  (Do)LOGBOOK

RELATED COMMANDS

LOOK displays the selection screens and is more useful if you are browsing.

DESCRIPTION

The commands (Do)LOGBOOK and (Do)LOOK do these things once:

• Saves the earlier screen.
• Defines the LOGBOOK and LOOK screens: The three Logbook screens are:
  • The setting and the topic buffer screen.
  • The browsing selection criteria screen.
  • The topic entry screen.
• Defines the Gold keys for these screens.
• Defines the HELP topics.

Each time this command is given it enables the data entry screens.

This command is automatically done for any (Do)SET or the (Do)MAKE commands. But until this happens the Gold keys and (Do)HELP do not work. Normally the Gold key is PF1 and is struck before its qualifier key. The Gold keys that are defined by Logbook work on the Logbook.

Note: The first operation on the database will take some time, please be patient with it. Each database operation in Logbook does an SQL COMMIT, but other programs can lock the database for extended periods forcing you to timeout.
Logbook Commands

LOOK

LOOK

Initializes the Logbook variables, keys, and help and displays the data selection screens for browsing.

FORMAT

(Do)LOOK

RELATED COMMANDS

LOGBOOK displays the settings and the date entry buffer and is more useful if you are entering new information. (Do)LOOK may be useful in selecting entries for (Do)EDIT ENTRY.

DESCRIPTION

The commands (Do)LOGBOOK and (Do)LOOK do these things once:

- Saves the earlier screen.
- Defines the LOGBOOK and LOOK screens: The two Look layouts are:
  - The setting and the topic buffer screen.
  - The selection criteria screen.
- Defines the Gold keys for these screens.
- Defines the HELP topics.

Each time this command is given it enables the browsing screens.

This command is automatically done for any (Do)SELECT or the (Do)SHOW commands that display the Logbook tables. But until this happens the Gold keys and (Do)HELP do not work. Normally the Gold key is PF1 and is struck before its qualifier key. The Gold keys that are defined by Logbook work on the Logbook.

Note: The first operation on the database will take some time, please be patient with it. Each database operation in Logbook does an SQL COMMIT, but other programs can lock the database for extended periods forcing you to timeout.
MAKE ENTRY

Creates a new entry (row) in the ENTRIES table, which is the holder of the comments made on shots, runs, and in general.

This (Do)MAKE command can be given by any user. New ENTRIES can be added as soon as the idea happens.

FORMAT (Do)MAKE ENTRY

SPEED-UP Gold-E

RELATED COMMANDS

(Do)SET TOPIC defines the TOPIC identifying each Logbook entry. Its speed-up is Gold-T.

(Do)SET SHOT defines the SHOT identifying each Logbook entry. Its speed-up is Gold-S.

(Do)SET RUN defines the RUN identifying each Logbook entry. Its speed-up is Gold-R.

(Do)SHOW VIEW displays any entries with the (Do)SET TOPIC, (Do)SET SHOT, and (Do)SET RUN settings. This speed-up is Gold-V.

(Do)SHOW ENTRIES displays the selected list of ENTRIES with their full TEXT.

DESCRIPTION

This command:

- Uses the (Do)SET TOPIC, (Do)SET SHOT, and (Do)SET RUN settings.
- Uses the LOGBOOK_topic buffer as the TEXT in for a new ENTRIES table entry.
- Requires the topic be valid. The topic is critical to searches.
- Requires the shot be valid unless the information is about the run or is general. Shot must be NULL then.
- Requires the run be valid unless the information is general. SHOT must be NULL then.
- Does not require entries be unique. Be neat with multiple entries. Use keywords to help searching.
**MAKE KEYWORD**

Creates a new entry (row) in the KEYWORDS table, which has a list of the suggested and approved words that can be used in creating entry TEXT in the ENTRIES table and in searching for the words.

**FORMAT**

```plaintext
(Do)MAKE KEYWORD  keyword brief
```

**ARGUMENTS**

- **keyword**
  Specifies the keyword (preferably without spaces) that will be used in adding to or searching for TEXT in the Logbook ENTRIES table.

- **brief**
  Specifies a short description of the keyword.

**prompts**

Keyword:
Briefly describe keyword keyword:

**restrictions**

This (Do)MAKE command can only be given by privileged users, typically the machine leader. New KEYWORDS are added when there are changes in the machine operation.

**RELATED COMMANDS**

- (Do)MAKE TOPIC adds a new TOPIC, which can be identified with a Logbook entry. The KEYWORDS should be "orthogonal" to the TOPICS.
- (Do)SHOW KEYWORDS displays the selected list of keywords and their brief descriptions. The KEYWORDS are all listed in the top buffer window after (Do)LOGBOOK or (Do)LOOK commands.
- (Do)SELECT KEYWORDS allows a list of keywords that will be searched for in the database on a (Do)SHOW command.

There may be about 30 keywords after the first year. These may be related to special conditions or setups for the machine.

**DESCRIPTION**

This command:
- Prompts for the keyword name. There is no default keyword.
- Prompts for a brief comment when not given.
- Requires keyword titles be unique and puts them in uppercase.

LOG–14
MAKE RUN

Creates a new entry (row) in the RUNS table, which has a list of the defined runs that are the collections of shots where machine parameters were similar to each other or are varied systematically. The run number in the ENTRIES table identifies these shots and can be used in searching for them.

FORMAT

(Do)MAKE RUN  run  brief

ARGUMENTS

run  Specifies the run number that will be identified with Logbook ENTRIES.

brief  Specifies a short description of the run.

prompts  Run:
            Briefly describe run number:

convention  The RUN numbers for Alcator-CMOD are numeric dates like 920123. For RFX, they are small sequential integers.

restrictions  This (Do)MAKE command can be given only by privileged users, typically the machine leader. New RUNS are added as the physics dictates.

RELATED COMMANDS

(Do)SHOW RUNS displays the selected list of runs and their brief descriptions.

(Do)SELECT RUNS allows a list of runs that will be searched for in the database on a (Do)SHOW command.

There may be about one run per day or two of machine operation.

(Do)SHOW VIEW displays any entries with the (Do)SET TOPIC, (Do)SET SHOT, and (Do)SET RUN settings.

DESCRIPTION

This command:

- Prompts for the run number. The default is the (Do)SET RUN value.
- The answer becomes the new (Do)SET RUN value.
- Prompts for a brief comment if none was given.
- Requires run numbers be unique.
MAKE SHOT

Creates a new entry (row) in the SHOTS table, which has a list of the defined shots. Each shot is a pulse of the machine with long times between shots compared to the recording time of the data. The shot number in the ENTRIES table identifies the comment with the shot and can be used in searching for the shot or a range of shots.

FORMAT 

(Do)MAKE SHOT  shot  run  brief

ARGUMENTS

**shot**
Specifies the shot number that will be identified with the Logbook ENTRIES table row. The default is the number given by the (Do)SET SHOT command. If the shot is found in the SHOTS table, then its run number will be set.

**run**
Specifies the run number with which to associate this shot. The default is the number given by the (Do)SET RUN command or found to match the shot number.

**brief**
Specifies a short description of the shot.

prompts

Shot (All,Null,NExt,Max,number)[set shot]:
Run (All,Null,NExt,Max,number)[set run]:
Briefly describe shot number:

conventions

The SHOT numbers for Alcator-CMOD are numeric dates with a three-digit shot of the day like 920123001. For RFX, they are small sequential integers.

restrictions

This (Do)MAKE command can be given only by privileged users, typically the machine control program. New shots are added as the machine takes data.

RELATED COMMANDS

(Do)SHOW SHOTS displays the selected list of shots with their brief descriptions. The highest shot number is shown in the top buffer window after (Do)LOGBOOK or (Do)LOOK commands.

(Do)SHOW VIEW displays any entries with the (Do)SET TOPIC, SHOT, RUN settings.

(Do)SELECT SHOTS allows a list of shots that will be searched for in the database on a (Do)SHOW command.

There may be 5 to 50 shots taken on a normal day of machine operation.
DESCRIPTION

This command:

- Prompts for the shot number. The default is the (Do)SET SHOT value.
- The answer becomes the new (Do)SET SHOT value. This would set the run number if it existed, which it should not.
- Prompts for the run number. The default is the (Do)SET RUN value.
- The answer becomes the new (Do)SET RUN value.
- Prompts for a brief comment if none were given.
- Requires shot numbers be unique.
MAKE TOPIC

Creates a new entry (row) in the TOPICS table, which has a list of the required subjects that can be used in creating entry text in the ENTRIES table and in searching for the subjects.

FORMAT

(Do)MAKE TOPIC  topic  brief

ARGUMENTS

**topic**
Specifies the topic (without spaces) that will be used by those making entries in the Logbook.

**brief**
Specifies a short description of the topic.

prompts

Topic:
Briefly describe topic name:

restrictions

This (Do)MAKE command can be given only by privileged users, typically the machine leader. New TOPICS are added when there are significant changes in the machine operation.

RELATED COMMANDS

(Do)MAKE KEYWORD adds a new KEYWORD, which will be used in the TEXT of a Logbook entry. The TOPICS should be "orthogonal" to the KEYWORDS.

(Do)SHOW TOPICS displays the selected list of topics with their brief descriptions. The TOPICS are all listed in the top buffer window after (Do)LOGBOOK or (Do)LOOK commands.

(Do)SHOW VIEW displays any entries with the (Do)SET TOPIC, (Do)SET SHOT, and (Do)SET RUN settings.

(Do)SELECT TOPICS allows a list of topics that will be searched for in the database on a (Do)SHOW command.

There may be about 30 topics after the first year, perhaps, one per diagnostic or machine control or group of diagnostics.

DESCRIPTION

This command:

• Prompts for the topic name. The default is the (Do)SET TOPIC value.
• The answer becomes the new (Do)SET TOPIC value.
• Prompts for a brief comment if none were given.
• Requires topic titles be unique and puts them in uppercase.
NEXT

Continues with the SCRIPT set by (Do)SCRIPT.

FORMAT  (Do)NEXT SCRIPT

DESCRIPTION  This command forces a continuation of the script of TPU commands begun by a (Do)SCRIPT command.

The next pause is at the end of the next script line.

This provides an escape from doing a MAKE ENTRY, otherwise required by the script.
Logbook Commands

NO

Puts back the screen that was present before the (Do)LOGBOOK or (Do)LOOK or any other of the commands that could invoke these were given.

FORMAT  

(Do)NO LOGBOOK

RELATED COMMANDS

LOGBOOK and LOOK set up the Logbook screens.

DESCRIPTION

The display set before the Logbook commands began is restored. That screen will not have the Gold keys but the (Do)HELP LOGBOOK command will still work.
PRINT

Writes a buffer to a file and prints and deletes it.

FORMAT

(Do)PRINT  [buffername]

ARGUMENTS

buffername
Specifies the name of a buffer to be printed.

prompts
Print which buffer (return cancels):

DESCRIPTION

This order of selection is followed:

• The buffername given.
• The section of the screen that is selected (highlighted).
• The prompted buffername.

If none of these condition is met, the command is not done. The buffer modification state is restored after the file is written as the temporary file SYS$SCRATCH:LOGBOOK.PRINT.
SCRIPT

Reads a file of TPU commands and executes them. Execution pauses at the end of each text line.

FORMAT

(Do)SCRIPT  filename

ARGUMENTS

filename

Specifies the name of a file with TPU commands, including logbook operations, to be interpreted in sequence.

prompts

Script filename:

RELATED COMMANDS

NEXT and MAKE ENTRY advance the script line.

DESCRIPTION

This command:

• Asks for the name of a file that has a list of TPU commands.
• Does an EXECUTE of the TPU command on each line of the file.
• Executes subsequent lines only when a (Do)MAKE ENTRY or a (Do)NEXT SCRIPT is given. The next line is then displayed and this continues until all are done.
• Allows any EVE/TPU command at the end of that command line, that is, after the script plays through the line.

EXAMPLE

A file "script." that loops to input comments on two topics and wait for an MDSplus event.

EVE_SET_SHOT("");EVE_SET_RUN("");EVE_SET_TOPIC("software")
EVE_SET_TOPIC("thomson")
x:=CALL_USER(1,"mdsshr,mds$wtevent,z");EVE_SCRIPT("script")

Note that there is no pause after the SET SHOT and SET RUN and they take their input from terminal for null arguments. The wait for event permits no TPU operations except type ahead. CALL_USER has problems with "non-standard" routines, like this one; it reports a returned value error.

Note: The full name of the Logbook or EVE commands must be given. Full names can be found by using $ SEARCH SYS$LIBRARY:LOGBOOK.TPU set_topic or another such command.
Quotes must be used around text arguments. The EVE_ prefix is required. Parentheses are required if there are any arguments, even optional ones. Semicolons separate multiple commands without pauses.

SCRIPT commands cannot be nested—the new one plays only. The SHOW buffer is overwritten.
SELECT ENTERED

Defines the list of ranges of dates entered that rows were added and are to be displayed by the (Do)SHOW commands.

FORMAT  

(Do)SELECT ENTERED  

criterion

ARGUMENTS  

criterion  

Specifies the list of dates of the original entries. Each list element is likely a range.

prompts  

Entered select(ALL,a:b like 25-DEC-1991 0:0,Y,TOD,TOM):
or Entered select(ALL,a:b ranges...; comma to add):

DESCRIPTION  

This command:

• Selects rows displayed by the (Do)SHOW commands.
• Can take an argument that is a special word
  • ALL for no selection.
  • NULL for those not voided.
  • NOT NULL for those voided.
  • that is a list of date ranges possibly with the special words YESTERDAY, TODAY, or TOMORROW. (Abbreviations are Y, TOD, TOM.) Each range has a colon separator.
• Prompts for the criterion if it is not given. A selection beginning with a comma is appended to the previous one.
• Affects (Do)SHOW RUNS, SHOTS, TOPICS, KEYWORDS, ENTRIES, that is, all tables.
• Messages the number of rows selected in the ENTRIES table.

EXAMPLE  

You can see yesterday’s entered shots.

(Do)Command:  SELECT ENTERED YESTERDAY:TODAY
(Do)Command:  SHOW SHOTS
SELECT KEYWORD

Defines the list of keywords in the text or brief comments displayed by (Do)SHOW commands.

FORMAT

(Do)SELECT KEYWORD  criterion

ARGUMENTS

criterion

Specifies the list of keywords of the entries.

prompts

Text select(ALL,list of string): or
Text select(ALL,list of string; comma to add):

DESCRIPTION

This command:

• Selects rows displayed by the (Do)SHOW commands.
• Can take an argument that is a special word
  • ALL for no selection.
  • that is a list of words, any of which will be found in the TEXT of the ENTRIES table or the BRIEF of the other tables.
• Prompts for the criterion if it is not given. A selection beginning with a comma is appended to the previous one.
• Affects (Do)SHOW RUNS, SHOTS, TOPICS, KEYWORDS, ENTRIES, that is, all tables.
• Messages the number of rows selected in the ENTRIES table.

EXAMPLE

You can see two keywords.

(Do)Command: SELECT KEYWORD Fluctuation, Equilibrium
(Do)Command: SHOW RUNS
SELECT ORDER

Defines the order of the rows displayed by (Do)SHOW commands.

FORMAT
(Do)SELECT ORDER  criterion

ARGUMENTS  criterion
Specifies the list of Table 1, Columns in the Tables to order (sort) the display of entries.

prompts
Order select(column list; full names): or
Order select(column list; full names; comma to add):

DESCRIPTION
This command:
• Selects the display listing order of entries that match the other selections and are displayed by the (Do)SHOW ENTRIES command.
• Can take an argument that is a special word
  • ALL for no ordering.
  • that is a list from the columns of the ENTRIES table: TOPIC, RUN, SHOT, ENTERED, USERNAME, TEXT, and VOIDED.
• Prompts for the criterion if it is not given. A selection beginning with a comma is appended to the previous one.
• Affects (Do)SHOW ENTRIES.
• Messages the number of rows selected in the ENTRIES table.

Note: RUNS are ordered by run number, SHOTS by shot number, TOPICS by topic name, and KEYWORDS by keyword.

EXAMPLE
You can see the entries ordered by shot and then topic.

(Do)Command: SELECT ORDER Shot, Topic
(Do)Command: SHOW ENTRIES
SELECT RUN

Defines the list of run numbers or ranges of runs displayed by (Do)SHOW commands.

FORMAT  (Do)SELECT RUN  criterion

ARGUMENTS  criterion
Specifies the list of run numbers. Each list element can be a range.

prompts
Run select(NULL,ALL,Max,Next, list or ranges): or
Run select(NULL,ALL,Max,Next, ...; comma to add):

DESCRIPTION
This command:
• Selects rows displayed by the (Do)SHOW commands.
• Can take an argument that is a special word
  • ALL for no selection.
  • NULL for those not defined, that is, general comments.
  • NOT NULL for those defined, that is, not general comments.
  • that is a list of number ranges possibly with the special words
    MAXIMUM or NEXT. (Abbreviations are M and NE.) Each range
    has a colon separator.
• Prompts for the criterion if it is not given. A selection beginning with
  a comma is appended to the previous one.
• Affects (Do)SHOW RUNS, SHOTS, ENTRIES.
• Messages the number of rows selected in the ENTRIES table.

EXAMPLE
You can see some runs.

(Do)Command: SELECT RUN 1,4:6
(Do)Command: SHOW SHOTS
Logbook Commands
SELECT SHOT

**SELECT SHOT**

Defines the list of shots numbers or ranges of shots displayed by (Do)SHOW commands.

**FORMAT**

(Do)SELECT SHOT  criterion

**ARGUMENTS**

*criterion*

Specifies the list of shot numbers. Each list element can be a range.

**prompts**

Shot select(NULL,ALL,Max,Next, list or ranges): or
Shot select(NULL,ALL,Max,Next, ...; comma to add):

**DESCRIPTION**

This command:

- Selects rows displayed by the (Do)SHOW commands.
- Can take an argument that is a special word
  - ALL for no selection.
  - NULL for those not defined, that is, run or general comments.
  - NOT NULL for those defined, that is, not a run or general comment.
  - that is a list of number ranges possibly with the special words MAXIMUM or NEXT. (Abbreviations are M and NE.) Each range has a colon separator.
- Prompts for the criterion if it is not given. A selection beginning with a comma is appended to the previous one.
- Affects (Do)SHOW SHOTS, ENTRIES.
- Messages the number of rows selected in the ENTRIES table.

**EXAMPLE**

You can see some shots.

(Do)Command: SELECT SHOT 101,104:106
(Do)Command: SHOW ENTRIES
SELECT TOPIC

Defines the list of topic names displayed by (Do)SHOW commands.

FORMAT

(Do)SELECT TOPIC  criterion

ARGUMENTS  criterion

Specifies the list of topic names. Each list element could be a range.

prompts

Topic select(ALL,from topic list): or
Topic select(ALL,from topic list; comma to add):

DESCRIPTION

This command:

• Selects rows displayed by the (Do)SHOW commands.
• Can take an argument that is a special word
  • ALL for no selection.
  • that is from the TOPICS table.
• Prompts for the criterion if it is not given. A selection beginning with
  a comma is appended to the previous one.
• Affects (Do)SHOW TOPICS, ENTRIES.
• Messages the number of rows selected in the ENTRIES table.

EXAMPLE

You can see two topics.

(Do)Command: SELECT TOPIC Software, MHD
(Do)Command: SHOW ENTRIES
SELECT USERNAME

Defines the list of usernames displayed by (Do)SHOW commands.

**FORMAT**

(Do)SELECT USERNAME  *criterion*

**ARGUMENTS**

*criterion*

Specifies the list of username names. Each list element can be a range.

**prompts**

Username select(ALL,User,list): or
Username select(ALL,User,list; comma to add):

**DESCRIPTION**

This command:

- Selects rows displayed by the (Do)SHOW commands.
- Can take an argument that is a special word
  - ALL for no selection.
  - USER for your own username.
  - that is the name of the user that created an entry. Each range has a colon separator.
- Prompts for the criterion if it is not given. A selection beginning with a comma is appended to the previous one.
- Affects (Do)SHOW RUNS, SHOTS, TOPICS, KEYWORDS, ENTRIES, that is, all tables.
- Messages the number of rows selected in the ENTRIES table.

**EXAMPLE**

You can display your own entries.

(Do)Command: SELECT USERNAME user
(Do)Command: SHOW ENTRIES
SELECT VOIDED

Defines the list of ranges of dates that entries were voided and are to be displayed by the (Do)SHOW ENTRIES command.

FORMAT

(Do)SELECT VOIDED criterion

ARGUMENTS

criterion

Specifies the list of dates the entries were voided. Each list element is likely a range.

prompts

Voided select(NULL, NOT null, ALL, a:b ranges of dates): or Voided select(NULL, NOT null, ALL, a:b ...; comma to add):

DESCRIPTION

This command:

• Selects rows displayed by the (Do)SHOW commands.
• Can take an argument that is a special word
  • ALL for no selection.
  • NULL for those not voided.
  • NOT NULL for those voided.
  • that is a list of date ranges possibly with the special words YESTERDAY, TODAY, or TOMORROW. (Abbreviations are Y, TOD, TOM.) Each range has a colon separator.
• Prompts for the criterion if it is not given. A selection beginning with a comma is appended to the previous one.
• Affects (Do)SHOW RUNS, SHOTS, TOPICS, KEYWORDS, ENTRIES, that is, all tables.
• Messages the number of rows selected in the ENTRIES table.

EXAMPLE

You can see yesterday’s voided shots.

(Do)Command: SELECT VOIDED YESTERDAY:TODAY
(Do)Command: SHOW ENTRIES
SET RUN

Defines the run number to use for (Do)MAKE commands and to be displayed by the (Do)SHOW VIEW command.

FORMAT  

(Do)SET RUN  criterion

prompts  Run (ALL,Null,NExt,Max,number)[previous]:

ARGUMENTS  criterion

Specifies the run number the new entry or is a special word:

- ALL does not set the run, views all runs. Use this only for very general comments about the machine.
- NULL does not set the run, views only NULLs. Use this only for very general comments about the machine.
- MAXIMUM for the largest run number entry.

speed-up  Gold-R

convention  The RUN numbers for Alcator-CMOD are numeric dates like 920123. For RFX, they are small sequential integers.

DESCRIPTION  This command:

- Sets run number of the (Do)MAKE ENTRY and (Do)SHOW VIEW commands.
- Prompts for the criterion if it is not given.
- Displays a "Use MAKE RUN" message if the run does not exist.
- Displays with the run number its BRIEF message.
- Affects the (Do)MAKE RUN, SHOT, ENTRY and (Do)SHOW VIEW commands.

Notes:  (Do)SET SHOT  looks up the run number for defined shots.
You cannot (Do)MAKE ENTRY until the shot is NULL or defined as part of a run.
EXAMPLE  You can show the entries with the highest run number.

(Do)Command: SET RUN MAX
(Do)Command: SHOW VIEW
Logbook Commands

SET SHOT

Defines the shot number to use for (Do)MAKE commands and to be displayed by the (Do)SHOW VIEW command.

FORMAT

(Do)SET SHOT  criterion

ARGUMENTS

criterion
Specifies the shot number the new entry or a keyword:

• ALL does not set the shot, views all shots. Use this only for general comments about the run.
• NULL does not set the shot, views only NULLs. Use this only for general comments about the run.
• MAXIMUM for the largest shot number entry.
• NEXT for the largest shot number entry plus one.

prompts
Shot (ALL,Null,NEXT,MAX,number)[previous]:

speed-up
Gold-S

convention
The SHOT numbers for Alcator-CMOD are numeric dates with a three-digit shot of the day like 920123001. For RFX, they are small sequential integers.

DESCRIPTION

This command:

• Sets shot number of the (Do)MAKE ENTRY and (Do)SHOW VIEW commands. This shot number will be used to look up the its run number. You will not need to do a (Do)SET RUN.
• Prompts for the criterion if it is not given.
• Displays with the shot number its BRIEF message.
• Displays a "Use MAKE SHOT" message if the shot does not exist.
• Affects the (Do)MAKE SHOT, ENTRY, and (Do)SHOW VIEW commands.

Note: (Do)SET SHOT looks up the run number for defined shots.

You cannot (Do)MAKE ENTRY until the shot is NULL or that defined as part of a run.

With caution, you can use the next highest shot. You must set the run manually.
EXAMPLE You can show the highest shot.

(Do)Command: SET SHOT MAX
(Do)Command: SHOW VIEW
SET TOPIC

Defines the topic name to use for (Do)MAKE commands and to be displayed by the (Do)SHOW VIEW command.

FORMAT  (Do)SET TOPIC  criterion

ARGUMENTS  criterion
            Specifies the topic name of the new entry.

prompts            Topic [previous]:

speed-up            Gold-T

DESCRIPTION  This command:
            • Sets topic name of the (Do)MAKE ENTRY and (Do)SHOW VIEW commands.
            • Prompts for the criterion if it is not given.
            • Displays a "Use MAKE TOPIC" message if the topic does not exist.
            • Affects the (Do)MAKE TOPIC, ENTRY and (Do)SHOW VIEW commands.
            • Is required for (Do)MAKE ENTRY.

EXAMPLE  You can show entries on a topic.

(Do)Command:  SET TOPIC mhd
(Do)Command:  SHOW VIEW
SHOW COLUMNS

Displays the columns of a table. For Logbook tables, see Table 1, Columns in the Tables.

FORMAT

(Do)SHOW COLUMNS  table

ARGUMENTS

table
Specifies the table or tables to use in the listing of the columns.

You can use SQL-style wildcards: % for any string and _ for one character.
(Compare to VMS wildcards of * and % for the same.)

prompts

Table name (wildcards: %=any and _=one):

RELATED COMMANDS

(Do)SHOW TABLES for the tables of the database.

DESCRIPTION

This command:

• Prompts for a table name, if not given.
• Shows the query and passes it to the SQL interpreter.
• Adds each row that matches the query to the SHOW buffer.
• Displays the table (relation) name, the column (field) name, the data type of the column, the length of the data item, and the scaling.
  (Scaling applies to decimal fractions stored as integers.)
• Returns to previous screens on next (Do) command.

Note: Rdb requires table and column names to be one word with uppercase alphanumeric plus dollar ($) and underscore (_) allowed.
SHOW ENTRIES

Displays the Logbook table named ENTRIES or portions of it. ENTRIES has the long comments on runs, shots, and machine conditions.

FORMAT

(Do)SHOW ENTRIES

RELATED COMMANDS

(Do)SELECT USERNAME, ENTERED, RUN, SHOT, TOPIC, VOIDED, and KEYWORD define the selection criteria.

(Do)SHOW VIEW shows portions of the ENTRIES table determined by the (Do)SET commands.

DESCRIPTION

This command:

• Builds a query from the AND of the selections for USERNAME, ENTERED, RUN, SHOT, TOPIC, VOIDED, and KEYWORD. Any element of a selection can satisfy that selection.

• Appends ordering by the ORDER selection. The default is TOPIC, RUN, SHOT, ENTERED, USERNAME. These are the allowed ORDER selections.

• Shows the query and passes it to the SQL interpreter.

• Adds each row of the ENTRIES table that answers the query to the SHOW buffer.

• Includes all columns from the ENTRIES table: ENTERED, USERNAME, RUN, SHOT, TEXT, and VOIDED.

• Returns to previous screens on next (Do) command.
SHOW GROUPS

Displays the groups of unique values of a column within a table. For Logbook tables, see Table 1, Columns in the Tables.

FORMAT

(Do)SHOW GROUPS  \textit{table column [where]}

ARGUMENTS

\textit{table}  
Specifies the table to use in the listing of the groups. No wildcards are allowed.

\textit{column}  
Specifies the column of the table to group on. No wildcards are allowed.

\textit{where}  
Optionally specifies further conditions for the lookup.

prompts

Table (list of tables):
Column (list of columns):
Where []:

RELATED COMMANDS

(Do)SHOW TABLES for the tables of the database.
(Do)SHOW COLUMNS for the columns of some tables.

DESCRIPTION

This command:
- Prompts for a table name displaying the full list of tables.
- Prompts for a column name displaying the full list of columns for the table chosen above.
- Prompts for any additional qualifiers for the where clause of the query.
- Shows the query and passes it to the SQL interpreter.
- Adds each row that matches the query to the SHOW buffer.
- Displays the number (count) of the distinct elements in the column of the table chosen.
- Returns to previous screens on next (Do) command.

Note: Rdb requires table and column names to be one word with uppercase alphanumeric plus dollar ($) and underscore (_@) allowed.
Logbook Commands
SHOW GROUPS

EXAMPLE
You can show how many rows of the ENTRIES table have different authors and the entry has not been marked as bad.

(Do)Command: SHOW GROUPS ENTRIES USERNAME "VOIDED IS NULL"
SHOW KEYWORDS

Displays the Logbook table KEYWORDS or portions of it. Keywords are words appearing in the TEXT of the ENTRIES table or BRIEF of the other Logbook tables. The KEYWORDS of an experiment are defined by the person in charge.

FORMAT (Do)SHOW KEYWORDS

RELATED COMMANDS (Do)SELECT USERNAME, (Do)SELECT ENTERED, and (Do)SELECT KEYWORD define the selection criteria.

DESCRIPTION This command:
- Builds a query from the AND of the selections for USERNAME, ENTERED, and KEYWORD. Any element of a selection can satisfy that selection.
- Appends ordering alphabetically by KEYWORD.
- Shows the query and passes it to the SQL interpreter.
- Adds each row of the KEYWORDS table that answers the query to the SHOW buffer.
- Includes columns from the KEYWORDS table: KEYWORD and BRIEF.
- Returns to previous screens on next (Do) command.
SHOW RUNS

Displays the Logbook table RUNS or portions of it. Runs are groups of shots with something in common.

FORMAT 

(Do)SHOW RUNS

RELATED COMMANDS

(Do)SELECT USERNAME, (Do)SELECT ENTERED, (Do)SELECT RUN, and (Do)SELECT KEYWORD define the selection criteria.

DESCRIPTION

This command:

• Builds a query from the AND of the selections for USERNAME, ENTERED, RUN, and KEYWORD. Any element of a selection can satisfy that selection.

• Appends ordering by RUN number.

• Shows the query and passes it to the SQL interpreter.

• Adds each row of the RUNS table that answers the query to the SHOW buffer.

• Includes RUN, ENTERED, and BRIEF columns from the RUNS table.

• Returns to the previous screens on next (Do) command.
SHOW SHOTS

Displays the Logbook table SHOTS or portions of it. A shot is a collection of data records taken essentially simultaneously.

FORMAT

(Do)SHOW SHOTS

RELATED COMMANDS

(Do)SELECT USERNAME, (Do)SELECT ENTERED, (Do)SELECT RUN, (Do)SELECT SHOT, and (Do)SELECT KEYWORD define the selection criteria.

DESCRIPTION

This command:

- Builds a query from the AND of the selections for USERNAME, ENTERED, RUN, SHOT, and KEYWORD. Any element of a selection can satisfy that selection.
- Appends ordering by SHOT number.
- Shows the query and passes it to the SQL interpreter.
- Adds each row of the SHOTS table that answers the query to the SHOW buffer.
- Includes SHOT, RUN, ENTERED, and BRIEF columns from the SHOTS table.
- Returns to the previous screens on next (Do) command.
SHOW TABLES

Displays the tables of the database. For the Logbook tables, see Section 4, Tables.

FORMAT

(Do)SHOW TABLES

RELATED COMMANDS

(Do)SHOW COLUMNS for the columns of some tables.

DESCRIPTION

This command:
• Shows the query and passes it to the SQL interpreter.
• Adds each row that matches the query to the SHOW buffer.
• Displays the table (relation) name, the number of rows (cardinality), and the view versus table indicator (flag). A view is a logical, not physical, table.
• Returns to previous screens on next (Do) command.

Note: Rdb requires table and column names to be one word with uppercase alphanumeric plus dollar ($) and underscore (_) allowed.
SHOW TOPICS

Displays the Logbook table TOPICS or portions of it. Topics are determined by the lead experimentalist and remain for the life of the experiment.

FORMAT

(Do)SHOW TOPICS

RELATED COMMANDS

(Do)SELECT USERNAME, (Do)SELECT ENTERED, (Do)SELECT TOPIC, and (Do)SELECT KEYWORD define the selection criteria.

DESCRIPTION

This command:

• Builds a query from the AND of the selections for USERNAME, ENTERED, TOPIC, and KEYWORD. Any element of a selection can satisfy that selection.

• Appends ordering alphabetically by TOPIC.

• Shows the query and passes it to the SQL interpreter.

• Adds each row of the TOPICS table that answers the query to the SHOW buffer.

• Includes TOPIC and BRIEF columns from the TOPICS table.

• Returns to previous screens on next (Do) command.
SHOW VIEW

Displays the portions of the Logbook table ENTRIES as selected by the (Do)SET commands.

FORMAT

(Do)SHOW VIEW

speed-up

Gold-V

RELATED COMMANDS

(Do)SET RUN, (Do)SET SHOT, and (Do)SET TOPIC define the view criteria. (Do)MAKE ENTRY to make the comment.

(Do)SHOW ENTRIES allows more restrictions on the selection of entries using the (Do)SELECT commands.

DESCRIPTION

This command:

• Builds a query from the AND of the settings for RUN, SHOT, and TOPIC.
• Appends ordering by the ENTERED date.
• Shows the query and passes it to the SQL interpreter.
• Adds each row of the VIEW table that answers the query to the SHOW buffer.
• Includes all columns from the ENTRIES table: ENTERED, USERNAME, RUN, SHOT, TEXT, and VOIDED.
• Returns to previous screens on next (Do) command.

Note: This one (Do)SHOW VIEW command uses the (Do)SET RUN, SHOT, TOPIC settings instead of the (Do)SELECT values. This allows you to preview or review comments made on the RUN, SHOT, or TOPIC that you are about to make a comment on or just finished making one on using (Do)MAKE ENTRY.
UNVOID

Confirms that you wish to mark as valid or good, the displayed selected ENTRIES that have been voided (marked as bad).

FORMAT

(Do)UNVOID [ENTRY]

prompts

UnVoid all these entries? (Yes/No)[N]

RELATED COMMANDS

(Do)VOID ENTRY to mark ENTRIES as bad.

DESCRIPTION

This command:

• Displays the selected entries with VOIDED IS NOT NULL.
• Confirms that you wish to UNVOID the selected entries.
• Marks each selected entry with NULL to unvoid it. The previous voiding date is removed.

Note: Generally, you will look only at VOIDED IS NULL entries.

You must be the "owner" of the comment.
### VOID

Confirms that you wish to mark as invalid or bad after displaying them, the selected ENTRIES that have not been voided (marked as bad).

### FORMAT

<table>
<thead>
<tr>
<th>(Do)VOID [ENTRY]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prompts</td>
</tr>
<tr>
<td>Void all these entries? (Yes/No)[N]</td>
</tr>
</tbody>
</table>

### RELATED COMMANDS

(Do)UNVOID ENTRY to mark ENTRIES as good.

### DESCRIPTION

This command:
- Displays the selected entries with VOIDED IS NULL.
- Confirms that you wish to VOID the selected entries.
- Marks each selected entry with the CURRENT_TIMESTAMP to void it.

**Note:** Generally, you will look only at VOIDED IS NULL entries, so this command will remove from the display these selected entries. You must be the "owner" of the comment.
Logbook Commands

**ZERO**

Clears the Logbook entry buffer for the currently (Do)SET TOPIC.

**FORMAT**

(Do)ZERO [ENTRY]

**prompts**

Zero buffer buffer name (Yes/No)[N]

**RELATED COMMANDS**

(Do)SET TOPIC to create the LOGBOOK_topic buffer.

**DESCRIPTION**

This command confirms that you wish to ZERO the buffer of the current topic.

**Note:** This could also be done by selecting the entire buffer and removing it, like:

(Gold)-5
(Select)
(Gold)-4
(Remove)
Requirements

Before you start, you or SYSTEM must do things found in:

- Section A.1, Quotas for system and user settings.
- Section A.2, Files for files needed to run.
- Section A.3, COM File for an alternative command file.
- Section A.4, Source code for where the sources are.

A.1 Quotas

You will often need some quota changes. Your SYSTEM person must do this and you must log in again. Some changes will be required to use DECcision; use the larger.

<table>
<thead>
<tr>
<th>Quota</th>
<th>Rdb</th>
<th>Decision</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASTLM</td>
<td>24</td>
<td>100</td>
<td>Asynchronous traps</td>
</tr>
<tr>
<td>BIOLM</td>
<td>18</td>
<td>n.a.</td>
<td>Block I/O limit</td>
</tr>
<tr>
<td>BYTLM</td>
<td>20480</td>
<td>30000</td>
<td>Bytes in IO transfer</td>
</tr>
<tr>
<td>DIOILM</td>
<td>18</td>
<td>n.a.</td>
<td>Direct I/O limit</td>
</tr>
<tr>
<td>ENQLM</td>
<td>2000</td>
<td>2000</td>
<td>Locks</td>
</tr>
<tr>
<td>FILLM</td>
<td>50</td>
<td>100</td>
<td>File limit</td>
</tr>
<tr>
<td>MAXDETACH</td>
<td>n.a.</td>
<td>0</td>
<td>Detached processes</td>
</tr>
<tr>
<td>PGFLQUO</td>
<td>20000</td>
<td>20000</td>
<td>Page file quota</td>
</tr>
<tr>
<td>WSEXENT</td>
<td>2048</td>
<td>n.a.</td>
<td>Working set extent</td>
</tr>
<tr>
<td>WSQUO</td>
<td>512</td>
<td>n.a.</td>
<td>Working set quota</td>
</tr>
</tbody>
</table>

A.2 Files

Then create the Logbook database once. A sample Structured Query Language (SQL) file is in HARPO::MDSSSOURCE: [KKLARE.LOGBOOK] LOGBOOK.SQL. For reference, put this file where you want the Logbook and define a temporary logical name $ DEFINE SQL$DATABASE disk:[area.LOGBOOK]LOGBOOK assuming you will want it on some disk or logical root in a directory. For now, this database should be in your local cluster. To create the database, key in $ MCR SQL$ @LOGBOOK.

You may want to change the GRANT operations in the LOGBOOK.SQL file before you start or the owner or SYSTEM can change them later.

The SYSTEM person must have copied:

- SYSSLIBRARY:TPUSQL.EXE
- SYSSLIBRARY:MDSSQLSHR.EXE
- SYSSLIBRARY:LOGBOOK.TPU
Requirements

- **SYSHELP:LOGBOOK.HLB** to your machine. These will be preferably in
- **MDSSROOT:[SYSLIB]**
- **MDSSROOT:[SYSHLP]**

See also Section A.3, COM File for personal definitions.

**Beware:** The TPU$CALLUSER logical will be different and incompatible with other TPU sections or command procedures. This will be fixed someday.

### A.3 COM File

If you do not have a handy **SYSTEM** person, use this alternative LOGBOOK.COM:

```plaintext
$ DEFINE TPU$CALLUSER disk:[area]TPUSQL/NOLOG
$ DEFINE SQL$DATABASE disk:[area.LOGBOOK]LOGBOOK/NOLOG
$ DEFINE/USER SYS$INPUT SYS$COMMAND
$ EDIT/TPU/COMM=disk:[area]LOGBOOK 'P1'
```

The help library is used inside **EVE**, also known as **EDIT/TPU**. To use it from the command line,

```plaintext
$ DEFINE HLP$LIBRARY SYS$LIBRARY:LOGBOOK
$ HELP @LOGBOOK
```

If for some reason the logical name TPU$SECTION is not defined as **EVE$SECTION** you will need to give this command line.

```plaintext
$ DEFINE TPU$SECTION EVE$SECTION
```

### A.4 Source code

The sources are built using a DEC/MMS "makefile" and uses MIT’s MDS$RULES.MMS from the MDS$DEVELOPMENT_ENVIRONMENT for the sources. They can be distributed by DEConet or magnetic tape.

MDS$SOURCE:[SQLSHR.CMSLIB] has the CMS sources for the Logbook. The source requires a Rdb development license for one module: MODULE.SQLMOD. Copying MODULE.MAR will allow use with the Rdb interactive license. A new database cannot be created under the run-time license. It might be copied (somehow) from another machine.

MDS$SOURCE:[LOG_MANUAL.CMSLIB] is the DEC/CMS documentation library and requires VAX DOCUMENT and its license to re-compose the manuals.
This SQL information is divided into several parts:

- Section B.1, Data Types about the forms of data.
- Section B.2, Expressions about combining data.
- Section B.3, Useful about the important operations.
- Section B.4, Seldom Used about less used operations.
- Section B.5, Nondynamic about internally used operations.

VAX/Rdb specific commands are marked with a number sign (#). No CDD or redundant VAX extensions are shown. Some standard conventions are used:

- CAPITALS are required words,
- lowercase is a name of something like a table or column or a value allowed as a parameter like 'text' or 123.
- [a] is optional value.
- [a|b] is an optional choice.
- {a|b} is a required choice.
- a,... is a repeatable, comma (or other) separated section.

## B.1 Data Types

The SQL data types are in Table B–1, SQL data types table.

<table>
<thead>
<tr>
<th>type</th>
<th>arg</th>
<th>default</th>
<th>alias maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHAR</td>
<td>(n)</td>
<td>1</td>
<td>16383</td>
</tr>
<tr>
<td>#VARCHAR</td>
<td>(n)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>#LONG VARCHAR</td>
<td>(n)</td>
<td>16383</td>
<td>16383</td>
</tr>
<tr>
<td>#TINYINT</td>
<td>(s)</td>
<td>DECIMAL/NUMERIC (1 to 4, s)</td>
<td></td>
</tr>
<tr>
<td>SMALLINT</td>
<td>(s)</td>
<td>DECIMAL/NUMERIC (5 to 9, s)</td>
<td></td>
</tr>
<tr>
<td>INTEGER</td>
<td>(s)</td>
<td>DECIMAL/NUMERIC (10 to 18, s)</td>
<td></td>
</tr>
<tr>
<td>#QUADWORD</td>
<td>(s)</td>
<td>DECIMAL/NUMERIC (19 and up, s)</td>
<td></td>
</tr>
<tr>
<td>REAL</td>
<td></td>
<td>FLOAT(1 to 24)</td>
<td></td>
</tr>
<tr>
<td>DOUBLE PRECISION</td>
<td></td>
<td>FLOAT(1 to 24)</td>
<td></td>
</tr>
<tr>
<td>#DATE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#LIST OF BYTE</td>
<td>(n)</td>
<td>1</td>
<td>65508</td>
</tr>
<tr>
<td>VARYING</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The argument n is the number of characters and the argument s is the power-of-ten scale factor.
Data values have these forms:

- 'String' is a text literal.
- "Double quoted string" is nonstandard text.
- Double the quoting character (’ or “) to include it.
- 123 or 1.2e3 are numeric literals.
- #1-May-1991 12:34:56.78' shows a string date
- # - 12:34' or 'TODAY' are other dates.
- NULL, USER, or #CURRENT_TIMESTAMP are keyword literals.

B.2 Expressions

These elements build the expressions you use in SQL. You should consult the "VAX Rdb/VMS SQL Reference Manual" and especially Part 1, Chapter 3, for a complete explanation of the terms.

```sql
expr =
{[+|-]
column
  [(SUM|AVG|MAX|MIN) (DISTINCT column|[ALL] expr)]
  COUNT(*|DISTINCT column)
  char_value
  (expr)
  # DBKEY
  # CURRENT_TIMESTAMP
) [+|-]*/...
| char-expr #([| char-expr ]... a pair of bars

char-expr =
{column|literal|USER|? a parameter marker
  #(column-select)
  SUBSTRING(char-expr FROM start [FOR length])
}

predicate = [NOT]
{expr [=|<|<=|>|>=|<>] expr
  expr [=|<|<=|>|>=] {ALL|ANY|SOME} (column-select)
  expr [NOT] BETWEEN expr AND expr
  # expr [NOT] CONTAINING expr case insensitive
  EXISTS(column-select)
  expr [NOT] IN (expr[|column-select],...)
  expr IS [NOT] NULL
  expr [NOT] LIKE pattern [ESCAPE char] %=any, _=one
  # expr [NOT] STARTING WITH expr
  # UNIQUE(column-select)
  |(predicate)
  } [AND|OR]...
Note: NULL (the missing marker) is not TRUE nor is it FALSE.

select-expr = SELECT [ALL|DISTINCT]
{[*|expr[#table|view|alias].*],...}
FROM {table|view} [alias],... [WHERE predicate]
[GROUP BY column,...] [HAVING predicate]
UNION [ALL] ... back to SELECT
```
column-select =
  { (column-select)
    | SELECT [ALL|DISTINCT] {*|expr|#(table|view|alias).*} \\
    | FROM {table|view} [alias],... [WHERE predicate] \\
    | [GROUP BY column,...] [HAVING predicate] \\
    | #UNION [ALL] ... back to SELECT
  }

B.3 Useful

These are possibly useful SQL commands. See also Section B.4, Seldom Used. ROLLBACK won't work in the Logbook because each command is COMMITed. You should consult the "VAX Rdb/VMS SQL Reference Manual" and especially Part 2, where the commands are diagrammed.

COMMIT [WORK]

CREATE VIEW view [(column,...)] AS select-expr \\
  # [ORDER BY {column|integer}[ASC|DESC], ...] \\
  # LIMIT TO limit ROWS
  [WITH CHECK OPTION #[CONSTRAINT constrain-name]]
#DECLARE [auth-id] SCHEMA [AUTHORIZATION]
FILENAME filename #RUNTIME FILENAME filename

GRANT
  |
  | { #[SELECT|INSERT|OPERATOR|DELETE|CREATETAB|ALTER}
  | [DROP|DBCTRL|DBADM|SHOW|REFERENCES|UPDATE]
  | [SECURITY|DISTRIBUTAN]
  |,...
  | ALL PRIVILEGES
  | ON SCHEMA AUTHORIZATION auth-id,...
  | { [SELECT|INSERT|DELETES#CREATETAB|ALTER}
  | [DROP|DBCTRL|SHOW|REFERENCES] [(column,...)]
  | [UPDATE] [(column,...)]
  |,...
  | ALL PRIVILEGES
  | ON [TABLE] {table|view},...
  | # (UPDATE|REFERENCES)...|ALL PRIVILEGES|ON COLUMN col,...
  | TO
  | { uic like ['*',']
  | #general-VMS-identifiers
  | #BATCH|NETWORK|INTERACTIVE|LOCAL|DIALUP|REMOTE
  |}...
  | PUBLIC
  | #AFTER {id|PUBLIC|POSITION n},... back to after TO

INSERT INTO {table|view|CURSOR cursor} [(column,...)]
  [VALUES (?|literal|NULL|USER|#CURRENT_TIMESTAMP),...]
  [#RETURNING DBKEY INTO parameter]
  select-expr
  )

UNION is VAXism

ROLLBACK [WORK]

SELECT select-expr FROM {table|view} [alias],...
  [WHERE predicate]
  [GROUP BY column,...] [HAVING predicate]
  UNION [ALL]
  [ORDER BY column]
  [LIMIT TO number ROWS]

UPDATE {table|view} [#{alias}] SET column = {expr|NULL},...
  WHERE [predicate|CURRENT OF cursor]
Occasionally Useful SQL commands. These are less used that the commands in Section B.3, Useful.

```sql
#ALTER {DOMAIN|INDEX|SCHEMA|TABLE} etc. to modify them.
#COMMENT ON {DOMAIN|COLUMN|TABLE|INDEX} name IS 'string'
#CREATE DOMAIN domain IS type
  [DEFAULT {literal|USER=NULL|CURRENT_TIMESTAMP}]
  [EDIT STRING IS 'string']
  [QUERY HEADER IS 'string'/...]
#CREATE [UNIQUE] INDEX index ON table (column [ASC|DESC]
  [SIZE IS n|MAPPING VALUES l to h],...)
  TYPE IS {HASHED|SORTED}
CREATE SCHEMA [AUTHORIZATION auth] #FILENAME file
  #CREATE DOMAIN etc.
  # CREATE INDEX etc.
  CREATE TABLE etc.
  # CREATE TRIGGER etc.
  CREATE VIEW etc.
  GRANT etc.

CREATE TABLE table {
  column
  {
    [data-type|#domain] [default]
    [PRIMARY KEY|NOT NULL|UNIQUE|CHECK (predicate)
    REFERENCES table [(column,...)]
  } [CONSTRAINT name]
  #|COMPUTED BY expression
  [CHECK (predicate)|NOT NULL] #|CONSTRAINT name
  |
  (PRIMARY KEY (column,...)
  UNIQUE (column,...)
  FOREIGN KEY (column,...)
  REFERENCES table [(column,...)]
  [CHECK(predicate)
  } #|CONSTRAINT name
  },...}

#CREATE TRIGGER trigger {BEFORE|AFTER}
  {INSERT|DELETE|UPDATE [OF column,...]
  ON table [REFERENCING {OLD|NEW} AS alias ...]
  WHEN predicate} {delete|update|insert|ERROR},...
  [FOR EACH ROW]

#DECLARE TRANSACTION
  [BATCH UPDATE
  [READ ONLY][READ WRITE] [WAIT wait|NOWAIT] [EVALUATING
  [auth-id] constraint AT [VERB|COMMIT] TIME, ...]
  [RESERVING {table|view}... FOR
  [EXCLUSIVE|PROTECTED|SHARED] [READ|WRITE]]
  |[ON auth-id,... USING (DEFAULTS|above stuff)] AND... ]

DELETE FROM {table|view} #[alias]
  WHERE (predicate|CURRENT OF cursor)

#DROP {CONSTRAINT constraint
  DOMAIN domain
  INDEX index
  SCHEMA [AUTHORIZATION auth-id|FILENAME file]
  TABLE table [CASCADE|RESTRICT]
  TRIGGER trigger
  DROP view [CASCADE|RESTRICT]
```
#EXPORT SCHEMA {AUTHORIZATION auth-id|FILENAME file} INTO file [WITH {EXTENSIONS|NOEXTENSIONS|DATA|NODATA}]

#IMPORT SCHEMA AUTHORIZATION auth-id FROM file [[NO]{ACL|BATCH UPDATE|CDD LINKS|DATA|TRACE}
FILENAME file
CREATE INDEX etc.
DROP INDEX etc.
#[USING {parameter,...|DESCRIPTOR descriptor}]

REVOKE etc. like GRANT with #ENTRY added to ALL PRIVILEGES.

#SET ALL CONSTRAINTS {ON|OFF} (Not usable in Logbook.)

#SET TRANSACTION etc. like DECLARE TRANSACTION

B.5 Nondynamic

These commands are use by dynamic SQL and not by the user.

CLOSE cursor

DECLARE cursor [INSERT ONLY|READ ONLY]
{CURSOR FOR select-expr
  [ORDER BY {column|integer}[ASC|DESC], ...]
  #[LIMIT TO limit ROWS]
  #[FOR UPDATE OF column,...]
  #[LIST CURSOR FOR SELECT column
    WHERE CURRENT OF table-cursor
  ]
} #DECLARE cursor [INSERT ONLY|READ ONLY] [LIST]
CURSOR FOR stmt_id

#EXECUTE IMMEDIATE statement

#FETCH cursor [INTO parameter,...|USING DESCRIPTOR descrip]

#FINISH

#DESCRIBE statement [SELECT LIST|MARKERS] INTO descriptor

OPEN {cursor}

#PREPARE {statement-id [SELECT LIST INTO descriptor]
  FROM statement-string

#RELEASE statement-id
**columns**: The units of information representing one instance of data placed in a database table. Each instance has the same data type and units.

**COMMIT**: An SQL term for updating of the database. Nothing is final until this is given.

**database**: An organized collection of data and, in particular, a relational database, where tables of information is in repeated rows of similarly structured columns.

**Do**: A key, (Do), on some keyboards that shows the EVE command line prompt so you can enter commands that do not have keys of their own.

**EVE**: The standard section file for TPU that allows screen editing like the older EDT program provided.

**Gold**: The key struck before another to change its meaning, thus doubling the number of keys. Usually set to PF1 on the keypad. It is not like the Shift key, which is held while the other key is struck.

**keypad**: The extra keys to the right of the typewriter keys that could be used for numeric entry or for special operation.

**keyword**: A word used to identify a subject within text.

**Logbook**: The program to access a database of text concerned with topics on runs, shots, and in general about experimental conditions.

**MDSplus**: A special collection of programs and libraries that controls an experiment and acquires, archives, and displays data.

**Rdb**: The relational database supported by Digital Equipment Corporation for VAX computers with concurrent, distributed access.

**ROLLBACK**: An SQL term for removing the changes since the last COMMIT.

**rows**: The collection of data representing one instance of the columns of a table.

**runs**: A collection of shots having some conditions in common. They may taken on the same day or over a number of days.

**section file**: A TPU term for a compile set of procedures save in a file.

**shots**: Machine bursts of information having many recordings. This pulse is separated from others by times long compared to the recording time.

**SQL**: The Structured Query Language (a standard with extensions) that accesses the database.
**Glossary**

**Tables:** An element of a database having rows repeating the columns of data. See also Section 4, Tables.

**Topics:** The subject matter of a data entry in the ENTRIES table, which is required to be the name of a row in the TOPICS table.

**TPU:** The standard VAX editor, Text Processing Unit, for character or window terminals.
Index

A

ALL • 6

B

book • 3
Browse Summary • 4
browsing • 5

C

CMS • A–2
columns • 5
Commands
CSQL • LOG–3
EDIT ENTRY • LOG–5
HELP LOGBOOK • LOG–6
INSERT ANSWER • LOG–7
INSERT QUERY • LOG–8
ISQL • LOG–9
LOGBOOK • LOG–11
LOOK • LOG–12
MAKE ENTRY • LOG–13
MAKE KEYWORD • LOG–14
MAKE RUN • LOG–15
MAKE SHOT • LOG–16
MAKE TOPIC • LOG–18
NEXT SCRIPT • LOG–19
NO LOGBOOK • LOG–20
PRINT • LOG–21
SCRIPT • LOG–22
SELECT ENTERED • LOG–24
SELECT KEYWORD • LOG–25
SELECT ORDER • LOG–26
SELECT RUN • LOG–27
SELECT SHOT • LOG–28
SELECT TOPIC • LOG–29
SELECT USERNAME • LOG–30
SELECT VOIDED • LOG–31
SET RUN • LOG–32
SET SHOT • LOG–34

Commands (cont’d)
SET TOPIC • LOG–36
SHOW COLUMNS • LOG–37
SHOW ENTRIES • LOG–38
SHOW GROUPS • LOG–39
SHOW KEYWORDS • LOG–41
SHOW RUNS • LOG–42
SHOW SHOTS • LOG–43
SHOW TABLES • LOG–44
SHOW TOPICS • LOG–45
SHOW VIEW • LOG–46
UNVOID ENTRY • LOG–47
VOID ENTRY • LOG–48
ZERO ENTRY • LOG–49
Command Section • LOG–3
COMMIT • 3
constraints • 5

D
database • 3
data entry layout • 5
data looking layout • 5
Descriptions • LOG–3
Do key • 4

E

EDT keypad • 4
Enabling Summary • 3
Entry making • 6

F

Files • A–1

G

General introduction • 3
<table>
<thead>
<tr>
<th>Index</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Summary</td>
<td>3</td>
</tr>
<tr>
<td>Gold key</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td></td>
</tr>
<tr>
<td>help</td>
<td>3</td>
</tr>
<tr>
<td>Hint</td>
<td></td>
</tr>
<tr>
<td>Bookreader</td>
<td>vii</td>
</tr>
<tr>
<td>Buffer review</td>
<td>3</td>
</tr>
<tr>
<td>Gold key</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>K</td>
<td></td>
</tr>
<tr>
<td>keypad</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>L</td>
<td></td>
</tr>
<tr>
<td>layouts</td>
<td>5</td>
</tr>
<tr>
<td>lists</td>
<td>6</td>
</tr>
<tr>
<td>LOGBOOK.COM</td>
<td></td>
</tr>
<tr>
<td>Alternative</td>
<td>A–2</td>
</tr>
<tr>
<td>Normal</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>M</td>
<td></td>
</tr>
<tr>
<td>MAXIMUM</td>
<td>6</td>
</tr>
<tr>
<td>MMS</td>
<td>A–2</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
</tr>
<tr>
<td>NEXT</td>
<td>6</td>
</tr>
<tr>
<td>Notation</td>
<td>4</td>
</tr>
<tr>
<td>NOT NULL</td>
<td>6</td>
</tr>
<tr>
<td>NULL</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td></td>
</tr>
<tr>
<td>PF1</td>
<td>4</td>
</tr>
<tr>
<td>PF2</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td></td>
</tr>
<tr>
<td>Quotas</td>
<td>A–1</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td></td>
</tr>
<tr>
<td>ranges</td>
<td>6</td>
</tr>
<tr>
<td>Rdb</td>
<td>3</td>
</tr>
<tr>
<td>Requirements</td>
<td>A–1</td>
</tr>
<tr>
<td>ROLLBACK</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td></td>
</tr>
<tr>
<td>Screens</td>
<td>5</td>
</tr>
<tr>
<td>Setup</td>
<td></td>
</tr>
<tr>
<td>Alternative</td>
<td>A–2</td>
</tr>
<tr>
<td>Normal</td>
<td>4</td>
</tr>
<tr>
<td>Source code</td>
<td>A–2</td>
</tr>
<tr>
<td>SQL</td>
<td>3</td>
</tr>
<tr>
<td>SQL Commands</td>
<td></td>
</tr>
<tr>
<td>Nondynamic</td>
<td>B–5</td>
</tr>
<tr>
<td>Seldom used</td>
<td>B–4</td>
</tr>
<tr>
<td>Useful</td>
<td>B–3</td>
</tr>
<tr>
<td>SQL Datatypes</td>
<td>B–1</td>
</tr>
<tr>
<td>SQL Expressions</td>
<td>B–2</td>
</tr>
<tr>
<td>SQL Information</td>
<td>B–1</td>
</tr>
<tr>
<td>SQL Summary</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Tables</td>
<td>4</td>
</tr>
<tr>
<td>Text Entry Summary</td>
<td>3</td>
</tr>
</tbody>
</table>