

# MDBS INTERACTIVE DATA MANIPULATION REFERENCE MANUAL

## The MDBS IDML MANUAL

Version 3.08

Micro Data Base Systems, Inc.

P. O. Box 248

Lafayette, Indiana 47902

USA

Telex: 209147 ISE UR

(312) 303-6300 (in Illinois)

December 1985

### Copyright Notice

This entire manual is provided for the use of the customer and the customer's employees. The entire contents have been copyrighted by Micro Data Base Systems, Inc., and reproduction by any means is prohibited except as permitted in a written agreement with Micro Data Base Systems, Inc.

## NEW RELEASES, VERSIONS, AND A WARNING

Any programming endeavor of the magnitude of the MDBS software will necessarily continue to evolve over time. Realizing this, Micro Data Base Systems, Inc., vows to provide its users with updates to this version for a nominal handling fee.

New versions of MDBS software will be considered as separate products. However, bona fide owners of previous versions are generally entitled to a preferential rate structure.

Finally, each copy of our software is personalized to identify the licensee. There are several levels of this personalization, some of which involve encryption methods guaranteed to be combinatorially difficult to decypher. Our products have been produced with a very substantial investment of capital and labor, to say nothing of the years of prior involvement in the data base management area by our principals. Accordingly, we are seriously concerned about any unauthorized copying of our products and will take any and all available legal action against illegal copying or distribution of our products.

# Table of Contents

|  | <u>Page</u> |
|--|-------------|
| CHAPTER I. OVERVIEW . . . . .                              | 1           |
| A. Introduction . . . . .                                  | 1           |
| B. Using IDML . . . . .                                    | 1           |
| C. IDML Commands . . . . .                                 | 2           |
| D. Notational Conventions . . . . .                        | 4           |
| E. On-Line Help . . . . .                                  | 4           |
| CHAPTER II. DISPLAY COMMAND . . . . .                      | 5           |
| A. Syntax . . . . .  | 5           |
| B. Synopsis . . . . .                                      | 5           |
| C. Description . . . . .                                   | 5           |
| D. Examples . . . . .                                      | 6           |
| CHAPTER III. INVOKING DML COMMANDS INTERACTIVELY . . . . . | 9           |
| A. Syntax . . . . .  | 9           |
| B. Synopsis . . . . .                                      | 9           |
| C. Description . . . . .                                   | 9           |
| D. Examples . . . . .                                      | 13          |
| CHAPTER IV. DEFINE COMMAND . . . . .                       | 15          |
| A. Syntax . . . . .  | 15          |
| B. Synopsis . . . . .                                      | 15          |
| C. Description . . . . .                                   | 15          |
| D. Examples . . . . .                                      | 17          |
| E. Advanced Usage of Macros with Parameters . . . . .      | 18          |
| CHAPTER V. READ, ECHO AND COMMENT COMMANDS . . . . .       | 19          |
| A. Syntax . . . . .  | 19          |
| B. Synopsis . . . . .                                      | 19          |
| C. Description . . . . .                                   | 19          |
| D. Examples . . . . .                                      | 20          |
| CHAPTER VI. QUIT COMMAND . . . . .                         | 21          |
| A. Syntax . . . . .  | 21          |
| B. Synopsis . . . . .                                      | 21          |
| C. Description . . . . .                                   | 21          |
| CHAPTER VII. STARTUP_FILE . . . . .                        | 23          |
| A. Overview . . . . .                                      | 23          |
| B. Creating the STARTUP File . . . . .                     | 23          |
| CHAPTER VIII. IDML ERROR MESSAGES . . . . .                | 25          |

## APPENDICES

|  |            |
|--|------------|
| APPENDIX A. DML Command Break-down by Function . . . . . | A-1 to A-3 |
| APPENDIX B. DML Commands . . . . .                       | B-1 to B-2 |
| APPENDIX C. DML Command String Forms . . . . .           | .C-1       |

## I. OVERVIEW

### A. Introduction

The Interactive Data Manipulation Language (IDML) is an approach to accessing MDBS III data bases. Alternative access approaches include the DML, QRS, IBS and RDL. IDML is similar to QRS in that it is interactive, independent of any host language, and supports several QRS commands (such as DISPLAY, READ, and user-defined macro commands). IDML is similar to the DML in that it supports most of the DML commands presented in the MDBS DMS Manual. With IDML, a user can execute each of these DML commands one-at-a-time. If a DML command requires data inputs, then the IDML processor prompts the user for appropriate data values. If a DML command extracts data from a data base, then the extracted values are displayed at the console.

IDML is particularly valuable as an aid in learning and testing DML logic. It is also valuable for quickly making low volume, ad hoc changes to data values and their interrelationships, without writing a program. All data integrity and security conditions as specified in the DDL are automatically enforced by IDML. Synonyms as defined with the DDL are generally recognized by IDML.

### B. Using IDML

Simple IDML installation and execution instructions are included in the pertinent system specific manual. Once IDML has been installed, it can be executed from the operating system. IDML can be invoked with a -b argument on the command line to explicitly allocate the page buffer region. If this argument is not specified, approximately half of the available memory is automatically allocated. The remainder of available memory is reserved as a non-data base working space (e.g., file control blocks, stack, sort work areas). If the argument is used, it has the form -bnnnnn where nnnnn is the (decimal) number of bytes to be allocated for the page buffer region. This number should be at least as large as the "minimum DMS buffer region size" reported by the DDL Analyzer, otherwise DMS command status error 31 results. If too large a page buffer region is requested, an error message indicating "excessive memory request" is displayed. The remaining memory reserved for non-data base working space is normally sufficient. However, in rare cases this working space may be insufficient, resulting in various kinds of error conditions and situations where the data base may be left open. In such a case, -b can be used to allocate a smaller page buffer region to allow a larger non-data base working space.

In the operating system command line, the user can also optionally specify -r followed by the name of the transaction log file to be used during the IDML execution (see the pertinent system specific manual). The disks containing the main data base area and the log file must be on-line.

When the IDML processor is executed, an IDML banner message appears on the console screen. This banner can be suppressed by including a -m argument in the command line used to invoke the IDML processor. In any event, the user is prompted for the file name of the main data base area, user name, and user password. These prompts can be suppressed by including -d, -u and -p arguments on the command line used to invoke IDML. If used, the -d should be followed immediately by the file name. Similarly, -u and -p should be followed by the user name and password respectively.

IDML checks to see whether the user name and password are valid for the indicated data base. If they are valid, then the I: prompt is issued and the data base is automatically opened. This means that IDML is ready to accept an IDML command. An IDML command is terminated by pressing the carriage return key. After IDML finishes its response to command, the I: prompt is again issued to show that IDML is ready to accept the next command.

Execution of IDML is terminated by entering either QUIT or BYE. This automatically closes the data base. This returns control to the operating system. Pressing the "soft interrupt" key (ESCAPE on most operating systems) interrupts the IDML activity and results in the I: prompt.

Chapter VII explains how to create and make use of a STARTUP file, which IDML can automatically utilize to identify the data base and user.

C. IDML Commands

The Interactive Data Manipulation Language supports a variety of standard commands. In addition, a user is allowed to define still other commands (called macro commands). The standard commands supported in IDML have the following effects:

- 1) HELP: provide on-line descriptions of IDML command usage
- 2) DISPLAY: display selected data dictionary contents
- 3) DML commands supported:

|         |       |       |        |       |       |
|---------|-------|-------|--------|-------|-------|
| AMM     | DRM   | FNOSK | GFM    | MRTF  | ONO   |
| AMO     | DRO   | FNS   | GFO    | M RTP | ONOI  |
| AOM     | FDRK  | FOI   | GMC    | MSF   | ONOSK |
| AOO     | FFM   | FOSK  | GOC    | MSP   | OOI   |
| AUI     | FFO   | FPM   | GTC    | NCI   | OOSK  |
| CCU     | FFS   | FPMI  | GTM    | ODRK  | OPM   |
| CRA     | FLM   | FPMSK | GTO    | OFM   | OPMI  |
| CRS     | FLO   | FPO   | IMS    | OFO   | OPMSK |
| DBCLSA  | FMI   | FPOI  | IOS    | OLM   | OPO   |
| DBCNV   | FMSK  | FPOSK | LGFLSH | OLO   | OPOI  |
| DBENV   | FNM   | FRK   | LGMSG  | OMI   | OPOSK |
| DBOPNA  | FNMI  | GETC  | MAU    | OMSK  | ORK   |
| DBSAVE  | FNMSK | GETM  | MCC    | ONM   | PFC   |
| DBSSTAT | FNO   | GETO  | MCF    | ONMI  | PFM   |
| DRC     | FNOI  | GFC   | MCP    | ONMSK | PFO   |

|      |     |     |     |     |     |
|------|-----|-----|-----|-----|-----|
| PUTC | SCD | SME | SOM | SUO | TOT |
| PUTM | SCM | SMM | SON | SUU | TUN |
| PUTO | SCN | SMN | SOO | TCN | XMM |
| RMS  | SCO | SMO | SOU | TCT | XMO |
| ROS  | SCU | SMU | SUC | TMN | XOM |
| RSM  | SDC | SOC | SUM | TMT | XOO |
| RSO  | SMC | SOE | SUN | TON |     |

These commands have the effects described in the MDBS DMS Manual. Multiuser DML commands will have no effect if a multiuser version of the IDML processor is not being used. Logging commands have no effect if the RTL form of IDML is not being used.

- 4) DEFINE: define macro commands
- 5) READ: read and execute the IDML commands on a disk file
- 6) ECHO: echo a message to the console screen
- 7) COMMENT: set the symbol used to denote a comment
- 8) FILLCHR: set the symbol to be used as a fill character during data entry
- 9) QUIT: stop processing IDML commands

These commands are described in the chapters that follow. The commands are presented in the order shown above. Figure I-1 shows the sample logical structure presented in Chapter II of the MDBS DDL Manual (in Figures II-3 and III-1,2). One additional set (ISKL) has been added to this schema. All examples in this manual are based on the logical structure of Figure I-1.

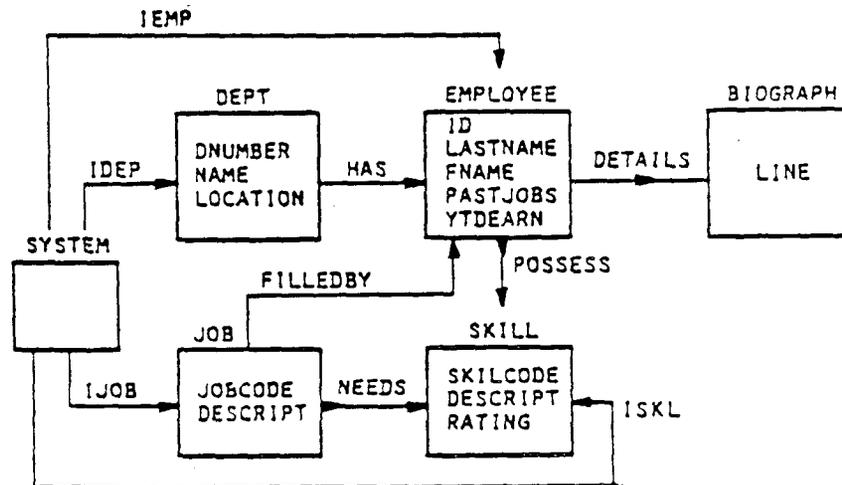


Figure I-1. Sample MDBS III Schema

## D. Notational Conventions

When describing the syntax of an IDML command, each clause is enclosed within the symbols < and >. These two symbols are not used to separate clauses within an IDML command itself. In actually stating an IDML command, two clauses in a command can be separated by either blanks or a comma.

For purposes of syntactic documentation, keywords and other literals are indicated with bold faced type. User-determined elements of a command are not in bold faced type. For example, the display command has the syntax:

**<display>****<parameter>****<list>**.

A display command must literally begin with the word **display**; the user determines the parameter and list that are to be used in a particular display command.

## E. On-Line Help

IDML provides an on-line help facility. To get a display of helpful information about IDML commands, type the word **help** in response to the I: prompt. Command syntax is displayed using notational conventions similar to those described above.

Command syntax for an individual DML command can be seen by typing the command name followed by a question mark (?) symbol.

## II. DISPLAY COMMAND

### A. Syntax

```

        <display>
          or
    <display><parameter><list>
          or
        <disp><parameter><list>
    
```

### B. Synopsis

A DISPLAY command can be used to obtain information from the data dictionary about areas, record types, sets, or data items. A DISPLAY command cannot be used to obtain data dictionary information if the data base was initialized with the o option (see MDBS DDL Manual, Chapter VI).

### C. Description

There are four permissible one-character parameters that can be used in a DISPLAY command:

| <u>&lt;parameter&gt;</u> | <u>information displayed about</u> |
|--------------------------|------------------------------------|
| a                        | all areas                          |
| i                        | all data items                     |
| l                        | language used for sorting          |
| r                        | all record types                   |
| s                        | all sets                           |

A list can optionally be specified with the a, i, r and s parameters in order to restrict the displayed information to a particular area, data item, record type, or set (respectively). This results in more detailed information than what is obtained without specifying the list.

| <u>&lt;parameter&gt;</u> | <u>&lt;list&gt;</u>              | <u>function</u>   |
|--------------------------|----------------------------------|---|
| a                        | area name                        | display information about the indicated area                    |
| i                        | record type name, data item name | display information about the item in the indicated record type |
| r                        | record type name                 | display information about the indicated record type             |
| s                        | set name                         | display information about the indicated set                     |

If both <parameter> and <list> are omitted from a DISPLAY command, the user is prompted with a list of the permissible parameters, as shown below:

The following display options are available:

- (a) display areas
- (i) display items
- (l) display language type
- (r) display records
- (s) display sets

Enter desired display option:

Virtually all schema information is available to a user through the DISPLAY command. The only kinds of data dictionary information that are not available with the DISPLAY command are user names, passwords, and user access codes. Read and write access codes for areas, data items, record types, or sets are not displayed. However, the user is informed whether he has read and/or write access to the area, data item, record type or set being displayed.

The Escape key can be used to interrupt the DISPLAY command's response as soon as the desired information has been displayed. The I: prompt reappears.

#### D. Examples

|                       |  |
|-----------------------|--|
| DISPLAY R             | displays summary information for all record types                                    |
| DISPLAY R SKILL       | displays detailed information for the SKILL record type                              |
| DISP S IEMP           | displays detailed information for the set IEMP                                       |
| DISP A                | displays summary information for all areas   |
| DISP I DEPT, LOCATION | displays detailed information for the LOCATION data item within the DEPT record type |
| DISP L                | displays the language used as a basis for sorting                                    |

Sample session using the DISPLAY command:

I: disp r

SYSTEM

title: (none)  
synonyms: (none)  
owned by: (none)  
owner of: \$SYSSET, IEMP, IDEP, IJOB  
items: (none)

DEPT

title: (none)  
synonyms: (none)  
owned by: IDEP  
owner of: HAS  
items: DNUMBER, NAME, LOCATION

EMPLOYEE

title: (none)  
synonyms: (none)  
owned by: IEMP, FILLEDBY, HAS  
owner of: POSSESS, DETAILS

BIOGRAPH

title: (none)  
synonyms: (none)  
owned by: DETAILS  
owner of: (none)  
items: LINE

SKILL

title: (none)  
synonyms: (none)  
owned by: POSSESS, NEEDS, ISKL  
owner of: (none)  
items: SKILCODE, DESCRIPT, RATING

JOB

title: (none)  
synonyms: (none)  
owned by: IJOB  
owner of: FILLEDBY, NEEDS  
items: JOBCODE, DESCRIPT

I: disp r skill

skill

```

title:          (none)
synonyms:       (none)
owned by:       POSSESS, NEEDS
owner of:       (none)
items:          SKILCODE UNSIGN      2
                 DESCRIPT STR      55
                 RATING REAL        2
    
```

I: disp s iemp

iemp

```

title:          (none)
synonyms:       (none)
type:           N:M
owner insertion: MANUAL
owner order:    IMMAT
owners:         SYSTEM
member insertion: AUTO
member order:   SORTED
members:        EMPLOYEE
    
```

I: disp a

JOB5

```

title:          (none)
synonyms:       (none)
access:         READ, WRITE
    
```

JOB1

```

title:          (none)
synonyms:       (none)
access:         READ, WRITE
    
```

JOB2

```

title:          (none)
synonyms:       (none)
access:         READ
    
```

I: disp i dept location

dept location

```

title:          (none)
synonyms:       (none)
type:           STR 35
    
```

### III. INVOKING DML COMMANDS INTERACTIVELY

#### A. Syntax

<dml command name><command string>

#### B. Synopsis

A DML command can be invoked interactively by stating its name and its command string. Just as with MDBS DMS, a command string consists of from zero to three arguments indicating the data item, record type, set and/or area names that are pertinent to the DML command with which the command string is being used (see Appendix C). However, in IDML some DML utility (e.g., AUI) and assignment (e.g., SUM) commands also require numbers in their command strings. If a DML command needs input data to store into the data base, then the user is prompted for the needed data. If a DML command extracts data from the data base, that data is displayed at the console.

A ? facility is provided to obtain the command string form for any DML command; it also indicates whether data will be passed to or from the data base. If a command status other than 0 results from executing a DML command, then a command status message is displayed at the console.

#### C. Description

To execute a DML command interactively, simply enter its name and an appropriate command string. DML command names that can be used within IDML are presented in Appendices A and B. The valid command string entries for each command are shown in Appendix C. A blank space or comma can be used to separate a command name from its command string. Entries within a command string are separated by either blanks or commas. When accepting a DML command, IDML translates the command string to upper case. Thus upper and lower case can be intermixed when stating a DML command string.

Command strings can contain area names, set names, data item names and numbers (i.e., integers). In addition to consulting Appendix C, a user can also find out about the command string form for a given DML command by entering the DML command name followed by ?. For instance, to find out the command string composition for SMM, enter:

```
smm?
```

IDML responds with:

```
syntax of SMM is: SMM set set
```

which indicates that two set names are needed in the SMM command string. To see the command string for SMU, enter:

smu?

IDML responds with

syntax of SMU is: SMU set number

indicating that the first entry in the command string is a set name and the second entry is a number (in this case the number of a user-defined currency indicator).

If a command involves data input to or data output from the data base, then the ? facility will show that a data transfer is involved. For instance,

crs?

will result in the following IDML response:

syntax of CRS is: CRS record [data]

showing that the command string consists of a record name and that a data transfer is involved. The "[ ]" surrounding the word "data" indicates that data is not entered on the command line. It is not part of the command string.

If IDML expects data input for a DML command, then the user is prompted for the needed data values. For example, to create an occurrence of the EMPLOYEE record type, the following interaction would be used (user responses are underlined).

```
I: crs,employee
   enter ID: 923487962
   enter LASTNAME: Smith
   enter FNAME: Bob
   enter PASTJOBS: Marketing
   enter PASTJOBS: Sales
   enter PASTJOBS:
   enter YTDEARN: 48000
```

The user terminates the entry of each data value by pressing the carriage return key. Notice that since PASTJOBS is a repeating data item (that occurs three times), the user is given three opportunities for specifying PASTJOBS values. In this example, the user specified only two of the three possible values.

When entering data for non-numeric data items, IDML reads in characters up to the maximum size of the data item with blank fill on the right. If something other than a blank is desired for character data items, the FILLCHR command can be used to designate a non-blank symbol as the fill character. When FILLCHR is invoked it should be

followed by either the desired symbol enclosed in a pair of quotes or an integer whose ASCII character equivalent will serve as the fill character. For instance,

```
FILLCHR 0
```

causes subsequent values entered for character data items to be filled with the null character. To later revert back to blank fill

```
FILLCHR " "
```

would be used.

When entering data for a numeric data item, IDML issues the DMS command status of 33 if the value is out of the maximum possible range allowed for that data item's size under IDML. If any value being entered is outside of the feasible range of values for the data item (if a feasibility range is specified in the DDL), then the DMS command status of 30 is returned. When prompted for a data value, pressing the carriage return key will be interpreted as a 0 for numeric data items and a blank for other data items.

If a non-zero command status results from interactively executing a DML command, then that command status is displayed at the console and the I: prompt is issued.

When data is retrieved by a command, that data is displayed at the console. For instance, to see the data that was just loaded by the above CRS command, the GETC command could be used:

```
I: getc
      ID: 923487962
      LASTNAME: Smith
      FNAME: Bob
      PASTJOBS: Marketing
      PASTJOBS: Sales
      PASTJOBS:
      YTDEARN: 48000
```

Notice that IDML provides the data item name for each data value.

Several utility commands (aui,ccu,dbenv), as well as assignment commands involving user-defined currency indicators (scu, smu, sou, suc, sum, sun, suo, suu), must have numbers specified in their command strings. With MDBS DMS these numbers were not part of command strings, but were passed with data blocks or program record types. The use of a number within a command string is illustrated by:

```
I: sum?
      syntax for SUM is: SUM number set
I: sum 7 set3
I:
```

The result is that user indicator 7 is assigned to be the same as the current member of SET3.

If a set is omitted from a command string, it defaults to the last set name specified in a command string. The same is true for number and area, data item, and record type names. As an example, suppose that we want to scroll through all department names:

```

I: ffm?
    syntax for FFM is: FFM set
I: gfm?
    syntax for GFM is: GFM item set [data]
I: ffm idep
I: gfm name
    NAME: cust support
I: fnm
I: gfm
    NAME: production
I: fnm
I: gfm
    NAME: R & D
.
.
.
I: fnm
    
```

DMS command status 255

Notice that in the first usage of GFM the set name was not specified, so it defaulted to the last set name that had previously been specified (i.e., idep). The same is true for the usage of FNM. In the second (and subsequent) usage of GFM the data item name was not specified, so it defaulted to the last item name that had previously been specified (i.e., name). In commands where two sets are required, the last set is the one saved. For example, if

som has idep

is executed, idep is the default set type remembered by IDML. The same convention also holds if two numbers are required for a command string.

D. Examples (user responses underlined)

```

I: crs?
    syntax for CRS is: CRS record [data]
I: crs dept
    enter DNUMBER: 14
    enter NAME: personnel
    enter LOCATION: Martinsville
I: ims?
    syntax for IMS is: IMS set
I: crs employee
    enter ID: 123456777
    enter LASTNAME: Moore
    enter FNAME: Steve
    enter PASTJOBS: clerk
    enter PASTJOBS:
    enter PASTJOBS:
    enter YTDEARN: 58392
I: ims has
I: crs employee
    enter ID: 923487962
    enter LASTNAME: Smith
    enter FNAME: Bob
    enter PASTJOBS: Marketing
    enter PASTJOBS: Sales
    enter PASTJOBS:
    enter PASTJOBS:
    enter YTDEARN: 48000
I: ims has
I: crs biograph
    enter LINE: Birthdate 3/21/28. Early years in Chicago area.
    enter LINE: Graduated from Purdue University 1949.
    enter LINE:
    enter LINE:
    enter LINE:
I: quit
    .
    .
    .
I: fmsk?
    syntax for FMSK is: FMSK set [data]
I: fmsk iemp
    enter LASTNAME: Moore
    enter FNAME: Bob

```

DMS command status 255

```
I: fmsk
      enter LASTNAME: Moore
      enter FNAME: Steve
I: gfc ytdearn
      YTDEARN: 58392
I: smm has iemp
I: geto has
      DNUMBER: 14
      NAME: personnel
      LOCATION: Martinsville
I: quit
```

## IV. DEFINE COMMAND

### A. Syntax

<define>

### B. Synopsis

The DEFINE command can be used to define macros (i.e., new, user-defined commands or parts of commands). These definitions are held in the data base and therefore exist across QRS and IDML sessions. The definition process itself is interactive and menu-driven.

A macro is a word that can be equated to any section of an IDML command and that can be freely substituted for that portion of the command. It can also be equated to an entire IDML command or to a sequence of IDML commands (involving many lines of input). Thus macros provide a convenient means for storing and later utilizing often-used IDML commands or sections of a command. A macro for a series of IDML commands is especially useful for iteratively executing the same DML logic. The user is prompted for any input needed by an IDML command within the macro. If a command within a macro returns command status of 255, the status is displayed and macro execution continues. Any other non-zero command status will cause macro termination.

A macro defined with IDML can later be used within QRS. A macro defined with QRS can later be used within IDML, provided its text consists of a valid command expression within IDML. Parameterized macros are supported in IDML. When a macro is to be used for any part of a DML command string, that macro's name must be in upper case.

Macros may not be nested within other macros. Macros may, however, be included within a command file executed via the READ command (see Chapter V). Because synonyms are treated very similarly to macros, a synonym will not be recognized within macro text.

### C. Description

When the DEFINE command is given, the following menu is displayed:

#### Macro Definition Utility

##### Macro/Synonym Functions:

- (A) Add a Macro
  - (C) Change a Macro
  - (D) Delete a Macro
  - (I) Index of Macro/Synonym Names
  - (L) List Macros and Synonyms
  - (?) Print This Command List
  - (Q) Quit Define Mode
- Function?

The effects of these seven macro definition functions are described below.

- A: The user is prompted for the name of the new macro and for the IDML text that it will represent. A macro name must be alphanumeric, and cannot begin with a digit. Use of this macro name within an IDML command will cause the macro's text to be substituted for the macro name. The text for a macro plus the length of the macro name cannot exceed 255 characters. Macros occurring within the text of another macro are not recognized. Synonyms occurring within the text of a macro are not recognized. The macro is not translated to upper case; it must be referenced in the same case with which it is defined in this function.
- C: The user is prompted for the name of an existing macro. The user is then prompted for the new text for that macro.
- D: The user is prompted for the name of the macro to be deleted.
- I: An index of all macros and synonyms is displayed. Synonyms are indented, macros are not.
- L: The user is prompted for a macro or synonym name. Pressing the return key will result in a listing of all macros (and their text) and all synonyms. If the user enters the name of a particular macro, its text is displayed. If the user enters the name of a particular synonym, then the name of the area, set, record type or data item for which it is a synonym is displayed. A user may not define a macro that has the same name as a synonym.
- ?: The macro definition functions are displayed.
- Q: The user leaves the macro definition mode. IDML gives the I: prompt.

D. Examples (underlined characters are entered by the user)

I: define

Macro Definition Utility

Macro/Synonym Functions:

- (A) Add a Macro
- (C) Change a Macro
- (D) Delete a Macro
- (I) Index of Macro/Synonym Names
- (L) List Macros and Synonyms
- (?) Print This Command List
- (Q) Quit Define Mode

Function?A

Name?CEMP

Enter text below (terminate with an empty line)

```
*fmi jobcode ijob
*som filledby ijob
*crs emp
*crs biograph
*fmi dnumber idep
*som has idep
*ims has
*
```

Function?A

Name?N

Enter text below (terminate with an empty line)

```
*fnm
*getc
*
```

Function?I

DNO

SC

CEMP

N

Function?C

Name?N

Enter text below (terminate with empty line)

```
*getc
*fnm
*
```

Function?Q

```

I: CEMP
   enter JOBCODE: 5
   enter ID: 123443210
   enter LASTNAME: Ferguson
   enter FNAME: Kristina
   enter PASTJOBS: sales rep
   enter PASTJOBS:
   enter PASTJOBS:
   enter YTDEARN: 29840
   enter LINE: Date of birth 12/13/51.
   enter LINE: Background in medicine and marketing.
   enter LINE: Salesperson of the year in 1979.
   enter LINE:
   enter LINE:
   enter DNUMBER: 7

```

```

I: CEMP

```

```

.
.
.

```

```

I: ffm idep

```

```

I: N

```

```

   DNUMBER: 3
   NAME: cust support
   LOCATION: Indianapolis

```

```

I: N

```

```

   DNUMBER: 5
   NAME: production
   LOCATION: Lafayette

```

```

I: N

```

```

.
.
.

```

```

I: N

```

```

   DNUMBER: 13
   NAME: sales
   LOCATION: Lafayette

```

DMS command status 255

### E. Advanced Usage of Macros with Parameters

When specifying the text of a macro with the DEFINE command, up to 9 parameters can be embedded within the macro text. Each macro parameter functions as a variable, whose value can be stated by a user whenever the macro is executed. For example, the text of a macro called SQ might be defined as:

```
SOM &1,&2
```

where &1 and &2 are used to denote macro parameters. If a user is interested in setting the current owner of POSSESS to be the same as the current member of HAS, then the macro would be executed as:

```
SQ(POSSESS,HAS)
```

If the user is later interested in setting the current owner of NEEDS to be the same as the current member of IJOB, then the same macro is executed, but with different parameter values:

```
SQ(NEEDS,IJOB)
```

As the above examples show, IDML substitutes the values specified by a user for the parameters declared in the original macro text. The first value appearing in the parentheses following the macro name is substituted for &1, the second value is substituted for &2, and so forth.

As a parameter is specified in macro text, it can be defined to be an optional parameter. When a user executes a macro, no value needs to be indicated for an optional parameter. A parameter is defined to be optional by appending the parameter with the = symbol followed by a space. For example, if the macro text for GP is

```
GETM &1
```

```
&2=
```

then the second parameter is optional. When executing the macro, a value for the second parameter may be omitted

```
GP(IJOB,)
```

or it may be included

```
GP(IJOB,PUTC)
```

Default values for parameters can also be specified in the macro text. For instance, if the macro text for SQ is

```
SOM &1, &2=HAS
```

then HAS is the default value for this macro's second parameter. Whenever a user executes SQ without specifying a value for the second parameter, IDML uses HAS as the value. For instance,

```
SQ(DETAILS,)
```

sets the current owner of DETAILS to be the same as the current member of HAS. If a user explicitly states a value for the second parameter, then IDML ignores the default value.

A parameter can be used more than once within a macro's text. As an example, the macro text

```
FMSK &1  
PFM &2,&1
```

finds a record and then gives one of its fields a new value. If the name of this macro is PMSK, then

```
PMSK (IEMP,LASTNAME)
```

finds a member of IEMP based on a sort key value and changes that member's last name.

Rules for Specifying Parameters in Macro Text

1. &1, &2, ....., &9 are valid macro parameters. If n parameters are specified, they should be &1, ... &n (where  $1 \leq n \leq 9$ ); &1 should be the leftmost parameter in the macro text; the remaining parameters can be specified in any order within the macro text.
2. Any parameter can appear more than once in a macro's text.
3. To signify that a parameter is optional, an equal sign and space must immediately follow the parameter number (e.g., &5= ) whenever it appears in the macro text.
4. To specify a default value for a parameter, the parameter number is immediately followed by an equal sign and the desired default value. Any letter or digit can appear in a default value. Any of the following characters is also allowed

" [ ] . / \* + - < > = # \$

No spaces, commas, or parentheses are allowed in the default value unless they are within a matching pair of double quotes. A parameter or macro cannot appear as part of a default value. If a parameter appears multiple times in a macro's text, it can have a different default value at each place it appears.

Rules for Executing a Parameterized Macro

1. The macro name is followed immediately (no space) by a matching set of parentheses containing the desired values. The values are separated by commas.
2. Macro evaluation is based on positional substitutions. IDML substitutes the first value in the parentheses for &1 (the leftmost parameter), the second value is substituted for &2 (the next parameter), and so forth. If no value is given for a parameter (i.e., there are two consecutive commas), then the parameter's default value is used. If it does not have a default value, but is optional, then the parameter is ignored. Omitting a value for a required parameter results in an error message.
3. The number of commas in a macro's list of values must always be one less than the number of parameters. If too many or too few values are specified, an error message results.
4. A parameter value cannot contain spaces, commas, or parentheses unless they are enclosed in a matching pair of double quotes. A macro cannot be used as a parameter value. In addition to digits and numbers, a value can contain any of the following characters:

" [ ] . / \* + - < > = # \$

## V. READ, ECHO AND COMMENT COMMANDS

### A. Syntax

```
<read><file-name><batch>
    <echo><message>
    <comment><symbol>
```

### B. Synopsis

The READ command causes IDML to read a disk file containing a series of IDML commands. IDML then executes all of these commands. The file that contains the IDML commands is indicated by the file-name clause. The batch option can be used to indicate that all input to the file's DML commands are contained in the file itself; user prompting will not occur.

An ECHO command causes a specified message to appear on the console screen. By embedding ECHO commands within the file processed by a READ command, an ongoing commentary of the READ command's processing action can be provided.

A COMMENT command is used to designate the symbol that will be interpreted as beginning a comment. Comments can be included for documentation purposes in a file processed by READ.

### C. Description

In using the READ command, the indicated file name must be quoted and must be fully qualified within the host operating system, including the drive number if it is not on the default drive. The file of IDML commands can be read, prepared, and/or saved through the same MDBS.DDL facilities that are used for creating, editing, and saving DDL specifications. Alternatively, any standard text editor can be used to create, read, and save the file of IDML commands. Any valid IDML command can appear in the READ command file, with the exception of a READ command. For example, the READ command can be used to execute a series of macros. If an error is encountered, the READ command is terminated and any remaining lines in the READ command file will not be read.

If the optional batch keyword is omitted, any DML command that requires user input results in user prompts. If batch processing is requested, each required input for a DML command will be read from succeeding lines in the file. Thus for batch processing, the file used by READ should contain data as well as DML commands. For example

```
CRS JOB
125
President
```

might be three lines in a READ command's file. Such a file must be processed in a batch fashion and will not prompt the user for data values. The above three lines would result in a new occurrence of JOB for a job whose code is 125 and whose description is President.

ECHO commands can be used in a READ command's file, in a startup file (Chapter VII) or in a macro's text (Chapter IV) to cause messages to appear on the console screen. The message following an echo keyword will be output to the next line on the screen. A message can be either a number or a string of characters enclosed in a matching pair of quotes. If there is no message, a blank line is output.

Whenever the ; symbol appears in a command, the remainder of the line is considered to be a comment and therefore is ignored by the IDML processor. The COMMENT command can be used to designate some other symbol as the indicator of comments. The symbol should be enclosed in quotes following the comment keyword.

#### D. Examples

```
READ "QFILE"
READ "B:BATC.H.TXT" BATCH
ECHO 29
ECHO "The instructions are as follows:"
ECHO
ECHO 'Processing is complete'
COMMENT ":"
```

## VI. QUIT COMMAND

### A. Syntax

<quit> or <bye>

### B. Synopsis

The QUIT (or BYE) command terminates an IDML session. It automatically closes the data base.

### C. Description

The quit command terminates IDML and returns the user to the operating system. QUIT closes the data base (saving all macro definitions in the main data base area). If a data base is not closed, it cannot be used in the future. The "soft interrupt" key (CONTROL-C on most operating systems) will abort any current IDML activity, but will not return control to the operating system. The IDML I: prompt results.

In addition to quit, bye may also be used to close the data base and return the user to the operating system.

Successful execution of the QUIT command is indicated by the "data base successfully closed" message.

This page intentionally left blank.

## VII. STARTUP FILE

### A. Overview

An IDML user can begin an IDML command session as described in Chapter I, Section B. Alternatively, a STARTUP file can be created which specifies the file name of the main data base area, user name, user password, and/or a series of IDML commands. When IDML is executed, its first activity is to read a file named STARTUP if one exists. If there is none, the IDML processor looks for a file named STARTI and treats it as the startup file if it exists. If the main area file name, user name, or user password is absent from the STARTUP file, the IDML user is prompted for the missing value(s). Then, all IDML commands on the STARTUP file are executed. When these commands have been executed, the IDML I: prompt is displayed.

### B. Creating the STARTUP File

The STARTUP file can be created, edited, and saved by using the MDBS.DDL software or any other standard text editing facilities. The approach used is similar to that used for creating an input file of IDML commands for the READ command (Chapter VI).

The first line of the STARTUP file must begin with the word START. A file name (for a data base's main area), user name and/or user password can optionally be specified on this line. If present, they must appear in this order. For example:

```
START "JOBS.DB" "D. LEHR" "smiles"
```

Here JOBS.DB is the fully qualified name of the file containing the main data base area, D LEHR is a user name, and "smiles" is the password. If any of these arguments is omitted from the first line of STARTUP, the user is prompted for the missing value(s). If the user fails to provide appropriate values, IDML returns control to the operating system. Each subsequent line of the STARTUP file can consist of any valid IDML command.

The STARTUP file must reside on the currently logged-on drive/directory and its name must be consistent with the operating system under which the IDML user is working.

In many operating systems, startup information can be created on files other than the STARTUP file. In this case, the startup information on such a file can be used in place of the STARTUP file by entering the file name on the operating system command line that is used to begin the execution of the IDML processor. See the pertinent system specific manual for details.

This page intentionally left blank.

## VIII. IDML ERROR MESSAGES

These are errors that can occur in the interactive utilization of IDML. If a DMS command is being executed with IDML, then various DMS command status messages can result. These are described in the MDBS DMS Manual, and will always be prefaced with the phrase "DMS command status". Those errors encountered within the IDML logic are described below.

\*\*\*Alternate input files may not be nested.

Explanation:

A file cannot be read from within another file which is being read.

Possible Causes and/or Solutions:

1. A 'READ' command has been used on a file which itself contains a 'READ' command.

\*\*\*Data dictionary not accessible.

Explanation:

The data dictionary cannot be accessed with the DISPLAY command.

Possible Causes and/or Solutions:

1. The data base was initialized with the o option.

\*\*\*Enclose file name in quotes.

Explanation:

Because of non-alphanumeric characters which are allowed for file names, for many operating systems, IDML requires that file names be in quotes.

Possible Causes and/or Solutions:

1. The file name specified must be enclosed in quotes.

\*\*\*File not present.

Explanation:

The file specified is not present on the disk.

Possible Causes and/or Solutions:

1. The system could not find the file indicated in the READ command.

**\*\*\*Insufficient room in memory.**

**Explanation:**

There is not enough room in memory to carry out any further processing by IDML.

**Possible Causes and/or Solutions:**

1. There is not enough room in memory to open the data base.
2. If the data base has already been opened, there is insufficient memory to process the given command.
3. See the appropriate system specific manual.

**\*\*\*Invalid area.**

**Explanation:**

Self-explanatory.

**Possible Causes and/or Solutions:**

1. The area name specified in an IDML command is not a valid area, as defined in the DDL specification.
2. Check list of valid areas using the DISP A command.

**\*\*\*Invalid DISPLAY parameter.**

**Explanation:**

The parameter specified is not a valid DISPLAY parameter.

**Possible Causes and/or Solutions:**

The parameter specified is not one of a, i, l, r, s.

**\*\*\*Invalid item name.**

**Explanation:**

The specified data item name does not exist for the record that is to be processed.

**Possible Causes and/or Solutions:**

1. The item name is misspelled or typed as record.item (which is invalid in IDML).
2. Check the command syntax for the proper sequence of arguments.

**\*\*\*Invalid number.**

**Explanation:**

The indicated field contains an invalid number.

**Possible Causes and/or Solutions:**

1. An invalid character has been specified within a numeric field.
2. Check numeric specifications in query command.

**\*\*\*Missing area.**

**Explanation:**

An expected area name is missing.

**Possible Causes and/or Solutions:**

1. An area was expected in the command string. It was missing and no area had been specified in any earlier IDML command.

**\*\*\*Missing item.**

## Explanation:

An expected data item name is missing.

## Possible Causes and/or Solutions:

1. A data item was expected in the command string. It was missing and no item name had been specified in any earlier IDML command.

**\*\*\*Missing number.**

## Explanation:

Self-explanatory.

## Possible Causes and/or Solutions:

1. A number was expected in the command string. It was missing and no number had been specified in any earlier IDML command.

**\*\*\*Missing record.**

## Explanation:

An expected record type name is missing.

## Possible Causes and/or Solutions:

1. A record type was expected in the command string. It was missing and no record type name had been specified in any earlier IDML command.

**\*\*\*Missing set.**

## Explanation:

An expected set name is missing.

## Possible Causes and/or Solutions:

1. A set was expected in the command string. It was missing and no set name had been specified in any earlier IDML command.

**\*\*\*Syntax error-command expected.**

## Explanation:

The system was expecting a command name and encountered a non-alphanumeric character within the command name.

## Possible Causes and/or Solutions:

1. A character within the command name was not alphanumeric.

**\*\*\*Unmatched quotes.**

## Explanation:

Self-explanatory.

## Possible Causes and/or Solutions:

1. Quotes have been improperly specified or improperly matched.

This page intentionally left blank.

**Appendix A**

**DML Command Break-down by Function**

Appendix A

page  
reference  
in MDBS DMS  
Manual

ASSIGNMENT COMMANDS

|     |   |      |
|-----|---|------|
| SCD | (Set Current of run unit to Data base key) . . . . .        | 40.1 |
| SCM | (Set Current of run unit based on Member) . . . . .         | 40.1 |
| SCN | (Set Current of run unit to Null) . . . . .                 | 40.2 |
| SCO | (Set Current of run unit based on Owner) . . . . .          | 40.2 |
| SCU | (Set Current of run unit based on User indicator) . . . . . | 41   |
| SDC | (Save Data base key for Current of run unit) . . . . .      | 41   |
| SMC | (Set Member based on Current of run unit) . . . . .         | 42   |
| SME | (Set Member to current of run unit (Exception)) . . . . .   | 43   |
| SMM | (Set Member based on Member) . . . . .                      | 43   |
| SMN | (Set Member to Null) . . . . .                              | 44   |
| SMO | (Set Member based on Owner) . . . . .                       | 44   |
| SMU | (Set Member based on User indicator) . . . . .              | 45   |
| SOC | (Set Owner based on Current of run unit) . . . . .          | 46   |
| SOE | (Set Owner to current of run unit (Exception)) . . . . .    | 46   |
| SOM | (Set Owner based on Member) . . . . .                       | 47   |
| SON | (Set Owner to Null) . . . . .                               | 47   |
| SOO | (Set Owner based on Owner) . . . . .                        | 48   |
| SOU | (Set Owner based on User indicator) . . . . .               | 49   |
| SUC | (Set User indicator to Current of run unit) . . . . .       | 49   |
| SUM | (Set User indicator to Member) . . . . .                    | 50   |
| SUN | (Set User indicator to Null) . . . . .                      | 50   |
| SUO | (Set User indicator to Owner) . . . . .                     | 51   |
| SUU | (Set User indicator to User indicator) . . . . .            | 52   |

BOOLEAN COMMANDS

|     |  |    |
|-----|--|----|
| AMM | (And of Members with Members) . . . . .  | 82 |
| AMO | (And of Members with Owners) . . . . .   | 83 |
| AOM | (And of Owners with Members) . . . . .   | 83 |
| AOO | (And of Owners with Owners) . . . . .    | 84 |
| XMM | (eXclude Members from Members) . . . . . | 85 |
| XMO | (eXclude Members from Owners) . . . . .  | 86 |
| XOM | (eXclude Owners from Members) . . . . .  | 87 |
| XOO | (eXclude Owners from Owners) . . . . .   | 88 |

CONNECT COMMANDS

|     |                                    |    |
|-----|------------------------------------|----|
| IMS | (Insert Member into Set) . . . . . | 57 |
| IOS | (Insert Owner into Set) . . . . .  | 58 |

CREATION COMMANDS

|     |                                     |    |
|-----|-------------------------------------|----|
| CRA | (Create Record in Area) . . . . .   | 54 |
| CRS | (Create Record and Store) . . . . . | 55 |

DELETION COMMANDS

|     |   |    |
|-----|---|----|
| DRC | (Delete Record that is <u>C</u> urrent) | 63 |
| DRM | (Delete Record that is <u>M</u> ember)  | 64 |
| DRO | (Delete Record that is <u>O</u> wner)   | 65 |

DISCONNECT COMMANDS

|     |  |    |
|-----|--|----|
| RMS | (Remove <u>M</u> ember from <u>S</u> et) | 59 |
| ROS | (Remove <u>O</u> wner from <u>S</u> et)  | 60 |
| RSM | (Remove all <u>S</u> et <u>M</u> embers) | 60 |
| RSO | (Remove all <u>S</u> et <u>O</u> wners)  | 61 |

FIND COMMANDS

|       |   |      |
|-------|---|------|
| FDRK  | (Find <u>D</u> uplicate Record based on calc <u>K</u> ey)             | 12   |
| FFM   | (Find <u>F</u> irst <u>M</u> ember)                                   | 13   |
| FFO   | (Find <u>F</u> irst <u>O</u> wner)                                    | 14   |
| FFS   | (Find <u>F</u> irst <u>S</u> equential record)                        | 14   |
| FLM   | (Find <u>L</u> ast <u>M</u> ember)                                    | 15   |
| FLO   | (Find <u>L</u> ast <u>O</u> wner)                                     | 15   |
| FMI   | (Find <u>M</u> ember based on data <u>I</u> tem)                      | 16   |
| FMSK  | (Find <u>M</u> ember based on <u>S</u> ort <u>K</u> ey)               | 17   |
| FNM   | (Find <u>N</u> ext <u>M</u> ember)                                    | 18   |
| FNMI  | (Find <u>N</u> ext <u>M</u> ember based on data <u>I</u> tem)         | 19   |
| FNMSK | (Find <u>N</u> ext <u>M</u> ember based on <u>S</u> ort <u>K</u> ey)  | 19   |
| FNO   | (Find <u>N</u> ext <u>O</u> wner)                                     | 20   |
| FNOI  | (Find <u>N</u> ext <u>O</u> wner based on data <u>I</u> tem)          | 21   |
| FNOSK | (Find <u>N</u> ext <u>O</u> wner based on <u>S</u> ort <u>K</u> ey)   | 22   |
| FNS   | (Find <u>N</u> ext <u>S</u> equential record)                         | 23   |
| FOI   | (Find <u>O</u> wner based on data <u>I</u> tem)                       | 24   |
| POSK  | (Find <u>O</u> wner based on <u>S</u> ort <u>K</u> ey)                | 25   |
| FPM   | (Find <u>P</u> rior <u>M</u> ember)                                   | 26   |
| FPMI  | (Find <u>P</u> rior <u>M</u> ember based on data <u>I</u> tem)        | 26   |
| FPMSK | (Find <u>P</u> rior <u>M</u> ember based on <u>S</u> ort <u>K</u> ey) | 27   |
| FPO   | (Find <u>P</u> rior <u>O</u> wner)                                    | 28   |
| FPOI  | (Find <u>P</u> rior <u>O</u> wner based on data <u>I</u> tem)         | 28.1 |
| FPOSK | (Find <u>P</u> rior <u>O</u> wner based on <u>S</u> ort <u>K</u> ey)  | 28.2 |
| FRK   | (Find Record based on calc <u>K</u> ey)                               | 28.3 |

MODIFY COMMANDS

|      |  |    |
|------|--|----|
| PFC  | (Put data into <u>F</u> ield of <u>C</u> urrent of run unit) | 34 |
| PFM  | (Put data into <u>F</u> ield of <u>M</u> ember)              | 34 |
| PFO  | (Put data into <u>F</u> ield of <u>O</u> wner)               | 35 |
| PUTC | ( <u>P</u> UT data into <u>C</u> urrent of run unit)         | 35 |
| PUTM | ( <u>P</u> UT data into <u>M</u> ember)                      | 36 |
| PUTO | ( <u>P</u> UT data into <u>O</u> wner)                       | 37 |

**RETRIEVAL COMMANDS**

|       |   |       |
|-------|---|-------|
| GETC  | (GET data from Current of run unit) . . . . .         | 30    |
| GETM  | (GET data from Member). . . . .                       | 30    |
| GETO  | (GET data from Owner) . . . . .                       | 31    |
| GFC   | (Get Field from Current of run unit). . . . .         | 31    |
| GFM   | (Get Field from Member) . . . . .                     | 32    |
| GFO   | (Get Field from Owner). . . . .                       | 32    |
| ODRK  | (Obtain Duplicate Record based on calc Key) . . . . . | 32.1  |
| OFM   | (Obtain First Member) . . . . .                       | 32.2  |
| OFO   | (Obtain First Owner). . . . .                         | 32.3  |
| OLM   | (Obtain Last Member). . . . .                         | 32.4  |
| OLO   | (Obtain Last Owner) . . . . .                         | 32.5  |
| OMI   | (Obtain Member based on data Item). . . . .           | 32.6  |
| OMSK  | (Obtain Member based on Sort Key) . . . . .           | 32.7  |
| ONM   | (Obtain Next Member). . . . .                         | 32.8  |
| ONMI  | (Obtain Next Member based on data Item) . . . . .     | 32.9  |
| ONMSK | (Obtain Next Member based on Sort Key). . . . .       | 32.10 |
| ONO   | (Obtain Next Owner) . . . . .                         | 32.11 |
| ONOI  | (Obtain Next Owner based on data Item). . . . .       | 32.12 |
| ONOSK | (Obtain Next Owner based on Sort Key) . . . . .       | 32.13 |
| OOI   | (Obtain Owner based on data Item) . . . . .           | 32.14 |
| OOSK  | (Obtain Owner based on Sort Key). . . . .             | 32.15 |
| OPM   | (Obtain Prior Member) . . . . .                       | 32.16 |
| OPMI  | (Obtain Prior Member based on data Item). . . . .     | 32.17 |
| OPMSK | (Obtain Prior Member based on Sort Key) . . . . .     | 32.18 |
| OPO   | (Obtain Prior Owner). . . . .                         | 32.19 |
| OPOI  | (Obtain Prior Owner based on data Item) . . . . .     | 32.20 |
| OPOSK | (Obtain Prior Owner based on Sort Key). . . . .       | 32.21 |
| ORK   | (Obtain Record based on calc Key) . . . . .           | 32.22 |

**UTILITY COMMANDS**

|        |   |      |
|--------|---|------|
| AUI    | (Allocate User Indicators). . . . .                         | 67   |
| CCU    | (Check Current of run unit against User indicator). . . . . | 68   |
| DBCLS  | (Data Base CLose) . . . . .                                 | 69   |
| DBCLSA | (Data Base CLose for Area). . . . .                         | 70   |
| DBCNV  | (Data Base format CONVersion) . . . . .                     | 70   |
| DBENV  | (Data Base ENVironment) . . . . .                           | 70.1 |
| DBOPNA | (Data Base OPeN Area) . . . . .                             | 72   |
| DBSAVE | (Data Base SAVE). . . . .                                   | 73   |
| DBSTAT | (Data Base STATistics). . . . .                             | 74   |
| GMC    | (Get Member Count). . . . .                                 | 75   |
| GOC    | (Get Owner Count) . . . . .                                 | 75   |
| GTC    | (Get Type of Current of run unit) . . . . .                 | 76   |
| GTM    | (Get Type of Member). . . . .                               | 76   |
| GTO    | (Get Type of Owner) . . . . .                               | 77   |
| NCI    | (Null all Currency Indicators). . . . .                     | 77   |
| TCN    | (Test Current of run unit for Null) . . . . .               | 78   |
| TCT    | (Test Current of run unit Type) . . . . .                   | 78   |
| TMT    | (Test Member Type). . . . .                                 | 78   |
| TOT    | (Test Owner Type) . . . . .                                 | 79   |

Appendix B

# DML Commands

Appendix B

|        | page<br>reference<br>in <u>MDBS DMS</u><br><u>Manual</u>     |
|--------|--|
| AMM    | (And of Members with Members) . . . . . 82                   |
| AMO    | (And of Members with Owners) . . . . . 83                    |
| AOM    | (And of Owners with Members) . . . . . 83-84                 |
| AOO    | (And of Owners with Owners) . . . . . 84-85                  |
| AUI    | (Allocate User Indicators) . . . . . 67-68                   |
| CCU    | (Check Current of run unit against User indicator) . . 68-69 |
| CRA    | (Create Record in Area) . . . . . 54-55                      |
| CRS    | (Create Record and Store) . . . . . 55-56                    |
| DBCLS  | (Data Base CLOSe) . . . . . 69                               |
| DBCLSA | (Data Base CLOSe for Area) . . . . . 70                      |
| DBENV  | (Data Base ENVironment) . . . . . 70-71                      |
| DBOPNA | (Data Base OPeN Area) . . . . . 72-73                        |
| DBSAVE | (Data Base SAVE) . . . . . 73-74                             |
| DBSTAT | (Data Base STATistics) . . . . . 74                          |
| DRC    | (Delete Record that is Current) . . . . . 63-64              |
| DRM    | (Delete Record that is Member) . . . . . 64                  |
| DRO    | (Delete Record that is Owner) . . . . . 65                   |
| FDRK   | (Find Duplicate Record based on calc Key) . . . . . 12-13    |
| FFM    | (Find First Member) . . . . . 13                             |
| FPO    | (Find First Owner) . . . . . 14                              |
| FPS    | (Find First Sequential record) . . . . . 14-15               |
| FLM    | (Find Last Member) . . . . . 15                              |
| FLO    | (Find Last Owner) . . . . . 15-16                            |
| FMI    | (Find Member based on data Item) . . . . . 16-17             |
| FMSK   | (Find Member based on Sort Key) . . . . . 17-18              |
| FNM    | (Find Next Member) . . . . . 18                              |
| FNMI   | (Find Next Member based on data Item) . . . . . 19           |
| FNMSK  | (Find Next Member based on Sort Key . . . . . 19-20          |
| FNO    | (Find Next Owner) . . . . . 20-21                            |
| FNOI   | (Find Next Owner based on data Item) . . . . . 21-22         |
| FNOSK  | (Find Next Owner based on Sort Key . . . . . 22-23           |
| FNS    | (Find Next Sequential record) . . . . . 23-24                |
| FOI    | (Find Owner based on data Item) . . . . . 24                 |
| FOSK   | (Find Owner based on Sort Key) . . . . . 25                  |
| FPM    | (Find Prior Member) . . . . . 26                             |
| FPO    | (Find Prior Owner) . . . . . 26-27                           |
| FRK    | (Find Record based on calc Key) . . . . . 27-28              |
| GETC   | (GET data from Current of run unit) . . . . . 30             |
| GETM   | (GET data from Member) . . . . . 30                          |
| GETO   | (GET data from Owner) . . . . . 31                           |
| GFC    | (Get Field from Current of run unit) . . . . . 31            |
| GFM    | (Get Field from Member) . . . . . 32                         |
| GFO    | (Get Field from Owner) . . . . . 32                          |
| GMC    | (Get Member Count) . . . . . 75                              |
| GOC    | (Get Owner Count) . . . . . 75                               |
| GTC    | (Get Type of Current of run unit) . . . . . 76               |
| GTM    | (Get Type of Member) . . . . . 76                            |
| GTO    | (Get Type of Owner) . . . . . 77                             |

|      | page<br>reference<br>in <u>MDBS DMS<br/>Manual</u>             |
|------|--|
| IMS  | (Insert Member into Set) . . . . . 57-58                       |
| IOS  | (Insert Owner into Set). . . . . 58                            |
| NCI  | (Null all Currency Indicators) . . . . . 77                    |
| PFC  | (Put data into Field of Current of run unit) . . . . . 34      |
| PFM  | (Put data into Field of Member). . . . . 34-35                 |
| PFO  | (Put data into Field of Owner) . . . . . 35                    |
| PUTC | (PUT data into Current of run unit). . . . . 35-36             |
| PUTM | (PUT data into Member) . . . . . 36                            |
| PUTO | (PUT data into Owner). . . . . 37                              |
| RMS  | (Remove Member from Set) . . . . . 59                          |
| ROS  | (Remove Owner from Set). . . . . 60                            |
| RSM  | (Remove all Set Members) . . . . . 60-61                       |
| RSO  | (Remove all Set Owners). . . . . 61                            |
| SCD  | (Set Current of run unit to Data base key) . . . . . 40.1      |
| SCM  | (Set Current of run unit based on Member). . . . . 40.1-40.2   |
| SCN  | (Set Current of run unit to Null). . . . . 40.2                |
| SCO  | (Set Current of run unit based on Owner) . . . . . 40.2        |
| SCU  | (Set Current of run unit based on User indicator). . . . . 41  |
| SDC  | (Save Data base key for Current of run unit) . . . . . 40.1    |
| SMC  | (Set Member based on Current of run unit). . . . . 42-43       |
| SME  | (Set Member to current of run unit (Exception)). . . . . 43    |
| SMM  | (Set Member based on Member) . . . . . 43-44                   |
| SMN  | (Set Member to Null) . . . . . 44                              |
| SMO  | (Set Member based on Owner). . . . . 44-45                     |
| SMU  | (Set Member based on User indicator) . . . . . 45              |
| SOC  | (Set Owner based on Current of run unit) . . . . . 46          |
| SOE  | (Set Owner to current of run unit (Exception)) . . . . . 46-47 |
| SOM  | (Set Owner based on Member). . . . . 47                        |
| SON  | (Set Owner to Null). . . . . 47-48                             |
| SOO  | (Set Owner based on Owner) . . . . . 48                        |
| SOU  | (Set Owner based on User indicator). . . . . 49                |
| SUC  | (Set User indicator to Current of run unit). . . . . 49-50     |
| SUM  | (Set User indicator to Member) . . . . . 50                    |
| SUN  | (Set User indicator to Null) . . . . . 50-51                   |
| SUO  | (Set User indicator to Owner). . . . . 51                      |
| SUU  | (Set User indicator to User indicator) . . . . . 52            |
| TCT  | (Test Current of run unit Type). . . . . 78                    |
| TMT  | (Test Member Type) . . . . . 78                                |
| TOT  | (Test Owner Type). . . . . 79                                  |
| XMM  | (eXclude Members from Members) . . . . . 85-86                 |
| XMO  | (eXclude Members from Owners). . . . . 86                      |
| XOM  | (eXclude Owners from Members). . . . . 87                      |
| XOO  | (eXclude Owners from Owners) . . . . . 88                      |

Appendix C

# DML Command String Forms

Appendix C

|                         |                       |
|-------------------------|-----------------------|
| amm, set, set, set      | ims, set              |
| amo, set, set, set      | ios, set              |
| aom, set, set, set      | lgflsh                |
| aoo, set, set, set      | lgmsg, data           |
| aui, number             | mau, data             |
| ccu, number             | mcc, data             |
| cra, record, area, data | mcf                   |
| crs, record, data       | mcp                   |
| dbclsa, area            | mrtf, record          |
| dbcnv, data             | mrtp, record          |
| dbenv, data             | msf, set              |
| dbopna, area, data      | mzp, set              |
| dbsave                  | nci                   |
| dbstat, data            | odr, record, data     |
| drc                     | ofm, set, data        |
| drm, set                | ofp, set, data        |
| dro, set                | olm, set, data        |
| fdrk, record, data      | olo, set, data        |
| ffm, set                | omi, item, set, data  |
| ffo, set                | omsk, set, data       |
| ffs, area               | onm, set, data        |
| flm, set                | onmi, item, set, data |
| flo, set                | onmsk, set, data      |
| fmi, item, set, data    | ono, set, data        |
| fmsk, set, data         | onoi, item, set, data |
| fnm, set                | onosk, set, data      |
| fnmi, item, set, data   | ooi, item, set, data  |
| fnmsk, set, data        | oosk, set, data       |
| fno, set                | opm, set, data        |
| fnoi, item, set, data   | opmi, item, set, data |
| fnosk, set, data        | opmsk, set, data      |
| fns, area               | opo, set, data        |
| foi, item, set, data    | opoi, item, set, data |
| fosc, set, data         | oposk, set, data      |
| fpm, set                | ork, record, data     |
| fpmi, item, set, data   | pfc, item, data       |
| fpmsk, set, data        | pfm, item, set, data  |
| fpo, set                | pfo, item, set, data  |
| fpoi, item, set, data   | putc, data            |
| fposk, set, data        | putm, set, data       |
| frk, record, data       | puto, set, data       |
| getc, data              | rms, set              |
| getm, set, data         | ros, set              |
| geto, set, data         | rsm, set              |
| gfc, item, data         | rso, set              |
| gfm, item, set, data    | scd, data             |
| gfo, item, set, data    | scm, set              |
| gmc, set, data          | scn, set              |
| goc, set, data          | sco, set              |
| gtc, data               | scu, number           |
| gtm, set, data          | sd, data              |
| gto, set, data          | smc, set              |

sme, set  
smm, set, set  
smn, set  
smo, set, set  
smu, set, number  
soc, set  
soe, set  
som, set, set  
son, set  
soo, set, set  
sou, set, number  
suc, number  
sum, set, number  
sun, number

suo, set, number  
suu, number, number  
tcn  
tct, record  
tmn, set  
tmt, record, set  
ton, set  
tot, record, set  
tun, number  
xmm, set, set, set  
xmo, set, set, set  
xom, set, set, set  
xoo, set, set, set

DOCUMENTATION COMMENT FORM

MDBS Document Title: \_\_\_\_\_

We welcome and appreciate all comments and suggestions that can help us to improve our manuals and products. Use this form to express your views concerning this manual.

Please do not use this form to report system problems or to request materials, etc. System problems should be reported to MDBS by phone or telex, or in a separate letter addressed to the attention of the technical support division. Requests for published materials should be addressed to the attention of the marketing division.

Sender:

|                    |             |
|--------------------|-------------|
| _____              | _____       |
| (name)             | (position)  |
| _____              | _____       |
| (company)          | (telephone) |
| _____              |             |
| (address)          |             |
| _____              |             |
| (city, state, zip) |             |

COMMENTS:

Areas of comment are general presentation, format, organization, completeness, clarity, accuracy, etc. If a comment applies to a specific page or pages, please cite the page number(s).

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Continue on additional pages, as needed. Thank you for your response.