

# MDBS RECOVERY & TRANSACTION LOGGING REFERENCE MANUAL

## The MDBS RTL 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. RTL Commands. . . . .	2
CHAPTER II. PAGE IMAGE POSTING . . . . .	3
A. Overview. . . . .	3
B. Command Details . . . . .	3
C. Example . . . . .	6
CHAPTER III. TRANSACTION LOGGING . . . . .	7
A. Overview. . . . .	7
B. Command Details . . . . .	7
C. Using RCV . . . . .	10
CHAPTER IV. TRANSACTION LOG FILE FORMAT . . . . .	15
CHAPTER V. RCV ERRORS . . . . .	17
CHAPTER VI. DMS COMMAND STATUS MESSAGES . . . . .	19

## I. OVERVIEW

### A. Introduction

A vital component of a data base management system is a facility for recovering from various catastrophic happenings such as media failure, system failure, or human error. The RTL form of MDBS has been designed to provide special recovery capabilities through its transaction logging and page image posting features. The RTL form consists of substantially modified data base control system, including a special group of DML commands for controlling the nature of transaction logging and page image posting that is desired during the execution of an application program. In some environments, the RTL form of MDBS III may require a larger page buffer than the standard form. Also supplied with the RTL form of MDBS is an interactive recovery utility called RCV.

A transaction has occurred whenever the data base contents are modified (creating, changing, or deleting data). RTL automatically logs all transactions made on a data base since the last data base back-up. When a media failure occurs, data base information on disk is destroyed. When this happens, RCV can be executed to automatically apply all logged transactions to the back-up copy of the data base. This restores the data base to a point close to the last completed transaction before the media failure occurred.

When a system failure (e.g., power interruption) occurs, information in the main memory buffers may be destroyed before the data base is properly closed. This means that the data base may be inconsistent, since some revised data base pages involved in a complex sequence of transactions may not yet have been flushed out to disk at the point of failure, while other revised pages involved in the complex transaction sequence have already been written to disk. RCV can be used to restore the data base to a consistent state. However, if a transaction was in the log file buffer at failure time, then that transaction is lost. Prior to invoking RCV it is prudent to make a back-up copy of the back-up copy of the data base. This can be time consuming for large data bases. Thus RTL provides an alternative defense against many system failures by allowing page image posting. This means that all pages revised during a complex transaction sequence are committed to disk only when that transaction is complete. System failure in the midst of a complex transaction will not result in data base inconsistency. Only the interrupted transaction has been lost when the data base is re-opened. Page image posting is not meaningful in multiuser processing situations, since all users share the same page buffer.

When human error occurs the data base has been changed in an authorized (from the standpoints of security and feasibility restrictions), but erroneous, manner. For instance, a data entry person may have entered transactions that were later discovered to be invalid. In such a situation it is important to be able to roll back the data base, effectively deleting the erroneous transactions. This selective restoration of a data base is accomplished with RCV, which allows the transaction log to be scanned to identify the erroneous

transactions. RCV is then used to selectively apply only the valid transactions on the log file to the data base back-up, with the result that the effects of invalid transactions have been eliminated from the data base.

**B. RTL Commands**

The RTL form of MDBS III supports all of the DML commands that are normally provided with the standard form. In addition, it offers seven other DML commands that are specifically related to recovery. Invoking a recovery command follows the same conventions as invoking any other DML command. The DML recovery commands fall into two groups:

- 1) those that deal with page image posting of complex transactions

PIFD	page image file declaration
TRBGN	begin posting a complex transaction sequence
TRCOM	the posted transaction sequence is committed to the data base
TRABT	abort the complex transaction sequence

- 2) those that deal with the logging of transactions

LGFILE	specify an alternative log file
TRBGN	start logging a complex transaction sequence
TRCOM	end the logging of a complex transaction sequence
LGFLSH	flush the log buffer to the log file
LGMSG	write a message to the log file

When using RTL, all transactions are logged; thus a log file must be on-line, preferably on some device other than the one(s) containing the data base. The use of a page image file is optional. RTL commands other than LGFLSH and LGMSG are not available for IDML processing. However, when IDML is configured with RTL (rather than DMS), automatic transaction logging of all IDML transactions occurs. The IDML user can specify the log file name on the operating system command line, as described in the appropriate system specific manual.

Chapter II describes page image posting. Chapter III describes transaction logging and the utilization of the RCV utility. The log file format is presented in Chapter IV. The final chapter explains the error messages that can be issued by the RCV recovery utility.



**Description**

The fully qualified file name, indicated by iblk's host language variable, is declared to be a page image preservation file for use by MDDBS.DMS in page image posting. If posting is desired, this command must be invoked prior to use of the TRBGN, TRCOM, and TRABT commands. The host language variable must be consistent with a string data item. The size (in bytes) is operating system dependent and is documented in the corresponding system specific manual. If PIFD is invoked, it must appear before the data base is opened.

**Examples of Command Usage**

```
block/direct    ... EO = PIFD ("iblk")
block/indirect  ... EO = DMS ("PIFD,iblk")
record/direct   ... EO = PIFD (iblk)
record/indirect ... EO = DMSD ("PIFD",iblk)
```

**TRABT**

**TR**ansaction **AB**ort

**TRABT**

**Command and Arguments**

TRABT (no arguments)

**Currency Indicators**

Used: none                      Changed: most currency indicators ← null

**Description**

**Posting Effects** (occur only if PIFD has been invoked):  
 This command aborts a complex transaction sequence that was initiated by the TRBGN command. A transaction cannot be aborted after it has already been committed (with the TRCOM command). This command invokes NCI internally to ensure that no inconsistent currency indicators will be present.

**Logging Effects:**

TRABT may be invoked only if PIFD has been invoked, in which case the transaction sequence that is aborted will be ignored by the RCV recovery processor.

**Examples of Command Usage**

```
block/direct    ... EO = TRABT (    )
block/indirect  ... EO = DMS ("TRABT")
record/direct   ... EO = TRABT (    )
record/indirect ... EO = DMS ("TRABT")
```

**TRBGN**

**TR**ansaction **Be**gin

**TRBGN**

**Command and Arguments**

TRBGN (no argument)

**Currency Indicators**

Used: none                      Changed: none

**Description**

**Posting Effects** (occur only if PIFD has been invoked):  
 This command denotes the beginning of a complex transaction sequence, involving the use of several DML commands. The changes that these DML commands cause in the data base do not become permanent until the TRCOM command is invoked. TRCOM commits these changes to the data base. The TRABT command aborts a complex transaction. After TRCOM or TRABT has been invoked, TRBGN must be re-invoked to initiate another complex transaction. If TRBGN is re-invoked without having issued a TRABT or TRCOM command, a command status error is issued and the re-invocation of TRBGN is ignored by MDBS.DMS.

**Logging Effects:**

When this command is invoked, it indicates the beginning of a complex sequence of transactions. Although all transactions are automatically logged, RCV will ignore any transactions that follow TRBGN unless a subsequent TRCOM command is encountered. The TRBGN and TRCOM commands are used in tandem to ensure that either all or none (if a complex sequence is not completed) of the transactions, which form part of a complex sequence, are used by RCV during recovery.

**Examples of Command Usage**

```
block/direct    ... EO = TRBGN (      )
block/indirect ... EO = DMS ("TRBGN")
record/direct  ... EO = TRBGN (      )
record/indirect... EO = DMS ("TRBGN")
```

**TRCOM**Transaction COMmit**TRCOM****Command and Arguments**

TRCOM (no arguments)

**Currency Indicators**

Used: none                      Changed: none

**Description**

**Posting Effects** (occur only if PIFD has been invoked):  
 This command commits a complex transaction, that was initiated by the TRBGN command, to the data base. It has the added effect of clearing the contents of the page-image file declared with the PIFD command.

**Logging Effects:**

This command indicates the end of a complex sequence of transactions. Any transactions logged since the last TRBGN command was invoked can now be processed by the RCV utility for data base restoration.

## Examples of Command Usage

```

block/direct    ... EQ = TRCOM (      )
block/indirect  ... EQ = DMS ("TRCOM")
record/direct   ... EQ = TRCOM (      )
record/indirect ... EQ = DMS ("TRCOM")

```

## C. Example

The following example illustrates a typical program flow when using page-image posting:

```

pifd
.
.
.
dbopn
.
.
.
ov (obtain and validate new data from end user)
.
.
.
trbgn
.
.
.
(incorporate new data into data base with appropriate DML
commands)
.
.
.
trcom
.
.
.
(iterate to ov, as desired)
dbcls

```

Notice that the page-image file is declared before opening the data base. The end user is prompted for new data. A complex transaction sequence is initiated by TRBGN. A sequence of transactions is generally considered to be complex if it involves related changes to several records. When TRCOM is executed, all changes that make up the complex transaction sequence are committed to the data base. If the program is interrupted after TRBGN and before TRCOM is completed, the changes are not committed to the data base. Even though the data base is not closed in this situation, it can be reopened and its contents are consistent.

### III. TRANSACTION LOGGING

#### A. Overview

Transaction logging allows an application program to make use of the log file defined in the DDL specification. The name of the file used for logging a program's transactions can be changed with the LGFILE command. All transactions since the last data base back-up are automatically logged onto the log file. The log file is used by the RCV utility (provided with MDBS-RTL). RCV can be used to automatically re-apply transactions to an old back-up copy of the data base, thereby recreating an up-to-date data base. The user has extensive control over which transactions are re-applied.

The TRBGN command is used to indicate the start of a complex sequence of transactions, for the purpose of transaction logging. The TRCOM command terminates the complex transaction sequence started by TRBGN. RCV will ignore any logged transactions that occur after an TRBGN, if a subsequent TRCOM was not reached during a run unit. Other commands that can be used in conjunction with transaction logging are LGFLSH and LGMSG which flush the log file buffer and insert messages into the log file, respectively.

The next section of this chapter describes the five recovery commands that can be used in conjunction with transaction logging. Section C explains how to use the RCV recovery utility to restore the data base or for the purpose of user surveillance.

#### B. Command Details

<b><u>TRBGN</u></b>	<b>TR</b> ansaction <b>Be</b> gin	<b><u>TRBGN</u></b>
---------------------	-----------------------------------	---------------------

Command and Arguments

TRBGN (no arguments)                      or                      LGCPLX (no arguments)

Currency Indicators

Used: none                                      Changed: none

Description

**Logging Effects:**

When this command is invoked, it indicates the beginning of a complex sequence of transactions. Although all transactions are automatically logged, RCV will ignore any transactions that follow TRBGN unless a subsequent TRCOM command is encountered. The TRBGN and TRCOM commands are used in tandem to ensure that either all or none (if a complex sequence is not completed) of the transactions, which form part of a complex sequence, are used by RCV during recovery.

**Posting Effects** (occur only if PIFD has been invoked):

This command denotes the beginning of a complex transaction sequence, involving the use of several DML commands. The changes that these DML commands cause in the data base do not become permanent until the TRCOM command is invoked. TRCOM commits these changes to the data base. The TRABT command aborts a complex transaction. After TRCOM or TRABT has been invoked, TRBGN must be re-invoked to initiate another complex transaction. If TRBGN is re-invoked without having issued a TRABT or TRCOM command, a command status error is issued and the re-invocation of TRBGN is ignored by MDBS.DMS.

**Examples of Command Usage**

```

block/direct    ... EO = TRBGN (      ) or EO = LGCPLX (      )
block/indirect  ... EO = DMS ("TRBGN") or EO = DMS ("LGCPLX")
record/direct   ... EO = TRBGN (      ) or EO = LGCPLX (      )
record/indirect ... EO = DMS ("TRBGN") or EO = DMS ("LGCPLX")
    
```

**TRCOM**

**Transaction COMMIT**

**TRCOM**

**Command and Arguments**

TRCOM (no arguments)            or            LGENDX (no arguments)

**Currency Indicators**

Used: none                    Changed: none

**Description**

**Logging Effects:**

This command indicates the end of a complex sequence of transactions. Any transactions logged since the last TRBGN command was invoked can now be processed by the RCV utility for data base restoration.

**Posting Effects** (occur only if PIFD has been invoked):

This command commits a complex transaction, that was initiated by the TRBGN command, to the data base. It has the added effect of clearing the contents of the page-image file declared with the PIFD command.

**Examples of Command Usage**

```

block/direct    ... EO = TRCOM (      ) or EO = LGENDX (      )
block/indirect  ... EO = DMS ("TRCOM") or EO = DMS ("LGENDX")
record/direct   ... EO = TRCOM (      ) or EO = LGENDX (      )
record/indirect ... EO = DMS ("TRCOM") or EO = DMS ("LGENDX")
    
```

**LGFILE**                                      **LoG FILE** specification                                      **LGFILE**

Command and Arguments

LGFILE,iblk

Currency indicators

Used: none                                      Changed: none

Description

The fully qualified name of a file to which transactions can be logged is specified by iblk's host language variable. This variable must be consistent with a string data item, whose maximum length is operating system dependent (see the appropriate system specific manual for size information). If the file to which transactions are to be logged is the same as the log file specified in the DDL specification, this command need not be invoked. If it is invoked, this command must appear before the DBOPN command.

Examples of Command Usage

```
block/direct      ... E0 = LGFILE ("iblk")
block/indirect   ... E0 = DMS ("LGFILE,iblk")
record/direct    ... E0 = LGFILE (iblk)
record/indirect  ... E0 = DMSD ("LGFILE",iblk)
```

**LGFLSH**                                      **LoG file buffer FLuSH**                                      **LGFLSH**

Command and Arguments

LGFLSH (no arguments)

Currency Indicators

Used: none                                      Changed: none

Description

The buffers holding transactions to be logged are flushed to the log file when this command is invoked. LGFLSH is automatically invoked by DBSAVE. This is especially useful when an important transaction has been processed by the system. It can be written to the log file before the log file buffer is full.

Examples of Command Usage

```
block/direct      ... E0 = LGFLSH (      )
block/indirect   ... E0 = DMS ("LGFLSH")
record/direct    ... E0 = LGFLSH (      )
record/indirect... E0 = DMS ("LGFLSH")
```

**LGMSG****LoG file MeSsaGe****LGMSG**

## Command and Arguments

LGMSG, iblk

## Currency Indicators

Used: noneChanged: none

## Description

The message specified in iblk's host language variable is written to the log file. This variable must be consistent with a string data item not exceeding 90 bytes. Messages written to the log file can be listed using the RCV utility program. The use of the LGMSG command is especially useful for surveillance.

## Examples of Command Usage

```
block/direct    ... E0 = LGMSG ("iblk")
block/indirect  ... E0 = DMS ("LGMSG, iblk")
record/direct   ... E0 = LGMSG (iblk)
record/indirect ... E0 = DMSD ("LGMSG", iblk)
```

### C. Using RCV

The RCV processor enables a user to display logged transactions and to apply these transactions to a copy of the back-up data base. RCV enforces all security conditions specified in the DDL. A user cannot display a transaction involving a data item, record type, set or area to which he does not have read access. A user cannot apply transactions to the data base unless he has write access to all data items, record types, sets, and areas involved in the transactions.

Before using RCV, several precautions should be taken:

**BACK UP YOUR DATA BASE BACK-UP!** If you are using RCV, it is because your master data base has been lost in some way. This means that the information in your back-up data base is very important. Do not risk losing it during the recovery process.

**BACK UP YOUR LOG FILE!** The RCV processor does not write to the transaction log file, but a back-up is cheap insurance.

RCV is executed from the operating system as described in the pertinent system specific manual. The disk containing the main data base area must be on-line. RCV 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). 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.

When RCV is executed, the RCV banner message appears on the console screen. This banner can be suppressed by including a -m argument in the command line used to invoke RCV. In any event, the user is prompted for the fully qualified 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 RCV. If used, the -d should be followed immediately by the main area's file name, the -u by the user name, and the -p by the password. RCV checks to see whether the user name and password are valid for the indicated data base.

If they are valid, the user is prompted to specify the log file name. If none is specified, the data base's default log file (as defined in the DDL specifications) is assumed. The log file name and all responses to subsequent RCV prompts can be pre-stored in a separate file. If this file name appears after a -i argument or the RCV command line, then its responses are automatically input in reaction to RCV prompts. Another optional argument is -v which means that RCV will operate in a verbose mode, giving a running commentary of its actions as it processes a log file.

Once a log file has been designated, a user is prompted with the following menu of options:

Options: (List, Quit, Reset, Scan, Update)?

After processing of a selected option is completed the menu of RCV options is again displayed. These options are described below.

1. Scan the log file (S)

When this option is selected, the user is prompted for the range of transactions to be scanned:

First transaction # [1]?  
 Last transaction # [highest]?

Transaction numbers are entered as eight hexadecimal digits (i.e., 00000001 through FFFFFFFF). Pressing a carriage return in response to the first transaction # has the same result as entering 00000001. Pressing a carriage return in response to the last transaction # has the same effect as entering the number of last transaction on the log file.

When the transaction range has been specified, the operator is presented with:

User name [all]?

This allows the operator to scan only that information which pertains to the specified user. The user name which is entered must be a name which appears in "User Login" transactions. If the default is entered (i.e., the RETURN or ENTER key is pressed), information pertaining to all users within the range is scanned.

In scanning the log file, RCV displays all messages, data base opens, data base closes, user logins, and user logouts appearing on the log file within the indicated range of transaction numbers. The user name associated with each login is also displayed.

2. List of the log file (L)

When this option is selected, the user is prompted for the range of transactions to be listed:

First transaction # [1]?  
 Last transaction # [highest]?

Transaction numbers are entered as eight hexadecimal digits (i.e., 00000001 through FFFFFFFF). Pressing a carriage return in response to the first transaction # has the same result as entering 00000001. Pressing a carriage return in response to the last transaction number has the same effect as entering the number of the last transaction on the log file.

When the transaction range has been specified, the operator is presented with:

User name [all]?

This allows the operator to list only that information which pertains to the specified user. The user name which is entered must be a name which appears in "User Login" transactions. If the default is entered (i.e., the RETURN of ENTER key is pressed), information pertaining to all users within the range is listed.

The list option displays all transactions in the log file within the indicated range of transaction numbers. The data values involved in transactions are not displayed by this command. This command is useful for determining the starting and ending transactions to be used in recovery processing.

The format of output from the list option is as follows:

```
run unit id transaction number .. transaction description
```

The transaction number is a unique eight digit hexadecimal number that identifies the transaction. The run unit id is a number assigned by RTL to a run unit; the meaning of this number is system specific and is generally of little interest to the RCV user. The following transaction descriptions can appear for a log file.

#### **Abort Transaction Sequence**

Appearance of this message indicates a successful invocation of TRABT in a source program. Any transactions between the preceding "Begin Transaction Sequence" and the abort will not be recovered.

#### **Begin Transaction Sequence**

Appearance of this message indicates that the user has called the TRBGN command in a source program. Any further transactions for this user will not be processed by RCV unless an TRCOM transaction or a user/system logoff transaction is encountered.

#### **Commit Transaction Sequence**

This transaction indicates the end of a transaction sequence.

#### **Create Record**

The record type of the record is listed, followed by the identifier of the newly created record. Note that if the record is the auto member or owner of any sets, several "Insert Record in Set" transactions may follow.

#### **Data Base Closed**

No additional information is listed. Note that for a multi-user system, this indicates the close of the data base. A run unit close is indicated by a User Logoff.

**Data Base Opened**

No additional information is listed.

**Delete Record**

The record identifier of the deleted record is listed.

**Insert Record in Set**

The name of the set is listed, followed by the identifier of the record which is being added to the set. For certain classes of sets, the identifier of the current owner (member) of the set is displayed, followed by the identifier of the current member (owner).

**Message**

A message entered in the transaction file by the LGMSG command is displayed.

**Put Data in Record**

The record type is displayed, followed by the record identifier of the record.

**Put Field in Record**

The record type is displayed, followed by the item type. The record identifier of the record to be stored will also be displayed.

**Remove Record From Set**

The name of the set, the current owner record's identifier and the current member's identifier are displayed.

**Remove Set Members**

The set name and its current owner identifier are displayed.

**Remove Set Owners**

The set name and its current owner identifier are displayed.

**User Login**

This transaction indicates that a specific user has been using the data base.

**User Logout**

This transaction indicates that a specific user has ended use of the data base.

### 3. Update the data base (U)

When this option is selected, the user is prompted for the range of transactions that are to be used for updating the copy of the back-up data base:

First transaction # [1]?  
Last transaction # [highest]?

Transaction numbers are entered as eight hexadecimal digits (i.e., 00000001 through FFFFFFFF). Pressing a carriage return in response to the first transaction # has the same result as entering 00000001. Pressing a carriage return in response to the last transaction number has the same effect as entering the number of the last transaction on the log file.

When the transaction range has been specified, the operator is presented with:

User name [all]?

This allows the operator to update using only those transactions in the range that pertain to the specified user. The user name which is entered must be a name which appears in "User Login" transactions. If the default is entered, transactions pertaining to all users within the range are used for updating.

The update option automatically applies all log file transactions within the indicated range for the indicated user(s) to the copy of the back-up data base.

### 4. Reset the log file (R)

When the reset option is selected, the user is prompted to assure that the reset is actually desired. The reset option has the effects of deleting all information from the indicated log file and re-initializing that file. **Extreme caution must be exercised when using this command, since it destroys the log file.** The user may wish to make an archival copy of the log file before the reset option is used.

If a data base has been initialized and loaded using MDBS DMS without RTL, that data base can later be used with RTL. When switching to RTL, execute the reset option first and revise any application programs that involve transactions. The necessary revision is that each program must make use of the LGFILE command, with the name of the file that was reset.

### 5. Quit (Q)

When this option is selected, the data base and log file are closed. Control is returned to the operating system.

#### IV. TRANSACTION LOG FILE FORMAT

Twenty functions are currently defined in the transaction file. A brief description of the format of each function is given on the following pages. The transaction file consists of a byte stream in which transactions immediately follow one another (i.e., no separator characters). The first byte of each logical disk record contains a hex A7, except that the last block contains a hex A8 value, indicating an end of file. The second byte is a checksum such that the sum of all bytes in the record is zero. There is an 'xx' byte header at the beginning of the file.

<u>Function</u>	<u>Value</u>	<u>Length</u>	<u>Meaning</u>	<u>Parameters(byte offset)</u>
NOP	0	1	No-op	none
CR	1	9+	Create Record	1 : Run Unit Id 2- 3: Data Length 4 : Record Type 5- 8: Record Identifier 9+ : Data Values
IMS	2	8	Add Record to Set (System-owned, non Next/Prior Set)	1 : Run Unit Id 2- 3: Set Type 4- 7: Record Identifier
IMS	3	12	Add Record to Set (System-Owned, Next/Prior Set)	1 : Run Unit Id 2- 3: Set Type 4- 7: Record Identifier 8-11: Adjacent Record Identifier
IMS	4	12	Add Record to Set (non System-Owned, non Next/Prior Set)	1 : Run Unit Id 2- 3: Set Type 4- 7: Record Identifier 8-11: Owner/Member Identifier
IMS	5	16	Add Record to Set (non System-Owned, Next/Prior Set)	1 : Run Unit Id 2- 3: Set Type 4- 7: Record Identifier 8-11: Owner/Member Identifier 12-15: Adjacent Record Identifier
IMS	6	16	Add Record to Set (non System-Owned, Next/Prior Set)  member->owner	1 : Run Unit Id 2- 3: Set Type 8-11: Owner/Member Identifier 12-15: Adjacent Record Identifier

MDBS RTL MANUAL - IV. TRANSACTION LOG FILE FORMAT - MDBS RTL MANUAL

<u>Function</u>	<u>Value</u>	<u>Length</u>	<u>Meaning</u>	<u>Parameters(byte_offset)</u>
PFC	7	11+	Set Field in Record	1 : Run Unit Id 2- 3: Data Length 4- 5: Item Type 6 : Record Type 7-10: Record Identifier 11+ : Data
PUTC	8	9+	Set Field in Record	1 : Run Unit Id 2- 3: Data Length 4 : Record Type 5- 8: Record Identifier 9+ : Data
RMS	9	12	Remove Member from Set	1 : Run Unit Id 2- 3: Set Type 4- 7: Owner Identifier 8-11: Member Identifier
RSM	10	8	Remove Set Members	1 : Run Unit Id 2- 3: Set Type 4- 7: Owner Identifier
RSO	11	8	Remove Set Owner	1 : Run Unit Id 2- 3: Set Type 4- 7: Member Identifier
DRC	12	6	Delete Record	1 : Run Unit Id 2- 5: Record Identifier
ULI	13	5+	User Login	1 : Run Unit Id 2- 4: External Run Unit Id 5+ : User Name
ULO	14	2	User Logout	1 : Run Unit Id
ESEQ	15	2	End Transaction Sequence	1 : Run Unit Id
BSEQ	16	2	Begin Transaction Sequence	1 : Run Unit Id
MSG	17	4+	Message	1 : Run Unit Id 2- 3: Message Length 4+ : Message
DBCL	18	1	Data Base Closed	none
DBOP	19	1	Data Base Opened	none
ABT	20	2	Abort Transaction Sequence	1 : Run Unit Id

## V. RCV ERRORS

**\*\*\* CAN'T CREATE FILE 'name'**

The specified file could not be created.

**\*\*\* CAN'T OPEN FILE 'name'**

The specified file could not be opened.

**\*\*\* DMS ERROR n DETECTED**

This message is usually caused by the same factors as the UPDATE ERROR FOUND message. Note that "n" corresponds to a data base error number.

**\*\*\* IMPROPERLY EMBEDDED TRANSACTION SEQUENCE**

Two TRBGN transactions were encountered without any intervening TRABT or TRCOM transactions.

**\*\*\* INSUFFICIENT MEMORY**

RCV does not have sufficient memory available to allocate needed buffers and tables. The pointers described in the installation information may be altered to increase the memory available.

**\*\*\* INVALID TRANSACTION TYPE ENCOUNTERED**

An invalid transaction was encountered in the transaction file. This is usually due to a damaged transaction file. An end of file is assumed at the error point in the file, and processing may continue.

**\*\*\* I/O ERROR ON FILE 'name'**

An Input/Output error was detected on the specified file.

**\*\*\* RUN UNIT ID NOT FOUND**

For multi-user systems, a compact user id is maintained in the transaction file. If a transaction with an invalid or unknown user id is encountered, this message is displayed. Usually this will only happen if a User Login function has not been encountered for the user, which is often the result of an improper specification of the first transaction number.

**\*\*\* SYSTEM CLOSE WITHOUT PREVIOUS OPEN**

This message may come about due to a damaged transaction file or due to use of a first transaction specification of a transaction which is not a "system open" transaction.

**\*\*\* SYSTEM OPENED BUT NOT PREVIOUSLY CLOSED**

This rather unusual message indicates that two "system open" transactions without an intervening "system close" transaction were encountered. This may indicate a damaged transaction file.

**\*\*\* UPDATE ERROR FOUND**

This message may indicate several problems. Generally, it occurs if the data base and transaction file do not correspond (i.e., using a transaction file associated with data base "A" won't work if you are applying it to data base "B").

**\*\*\* USE HEX NOTATION**

This message is generated when the user fails to enter a valid hex number for a RCV prompt.

**\*\*\* USER LOGIN NOT PREVIOUSLY LOGGED**

In a multi-user system, two "user login" transactions have been encountered without an intervening "user logout" transaction.

**\*\*\* USER TABLE OVERFLOW**

In a multi-user system, the transaction file may contain transactions from a large number of simultaneous users. Since RCV is initialized to handle 20 simultaneous users, this message probably indicates a damaged transaction file.

## VI. DMS COMMAND STATUS MESSAGES

There are several command status error messages that are exclusively related to the posting and logging as described in Chapters II and III:

**\*\*55\*\*** Log file I/O error **\*\*55\*\***

Possible Causes:

1. An I/O error was encountered when trying to open the log file.
2. See also Command Status 51 (in the MDBS DMS Manual).

**\*\*56\*\*** Log file not current **\*\*56\*\***

Possible Causes:

1. The log file is not up-to-date.

**\*\*57\*\*** Logging inhibited **\*\*57\*\***

Possible Causes:

1. An I/O error was detected on the log file.
2. The log file may already be open.

**\*\*58\*\*** Page-image file I/O error **\*\*58\*\***

Possible Causes:

1. An I/O error was encountered when an attempt was made to read from or write to the page-image preservation file.
2. See also Command Status 51 (in the MDBS DMS Manual).

**\*\*70\*\*** No transaction in progress **\*\*70\*\***

Possible Causes:

1. An attempt was made to abort or commit a transaction sequence without having previously invoked TRBGN.

**\*\*71\*\*** Transaction already in progress **\*\*71\*\***

Possible Causes:

1. TRBGN was invoked twice without having invoked TRABT or TRCOM between the two TRBGNs.

**\*\*72\*\*** Page-image posting not active **\*\*72\*\***

Possible Causes:

1. TRABT was invoked without having previously declared a page-image preservation file with PIFD.

**\*\*73\*\*** Page-image file index overflow **\*\*73\*\***

Possible Causes:

1. A transaction sequence was too long.
2. More than sixty pages have been altered in the course of a transaction sequence.

**\*\*74\*\*** Log file not present **\*\*74\*\***

Possible Causes:

1. Using the RTL form of MDBS, the data base was opened without the log file being present. Processing continues.

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.