

ATTN: CHARLIE GIBBS

01113
CAV208M45541 UP 8748

R1A

UAS

SPERRY UNIVAC
1 - 1818 CORNWALL STREET
VANCOUVER B C

V6J 1C7

**PUBLICATIONS
UPDATE**

Operating System/3 (OS/3)

IMS/DMS Interface

User Guide

This Library Memo announces the release and availability of Updating Package A to "SPERRY UNIVAC Operating System/3 (OS/3) IMS/DMS Interface User Guide", UP-8748 Rev. 1.

This update describes the following features for release 8.0:

- DMS support of multiple IMS systems
- Shutting down DMS before IMS
- Rollback error codes returned to IMS by DMS
- Support for multithread IMS in System 80

Copies of Updating Package A are now available for requisitioning. Either the updating package only, or the complete manual with the updating package may be requisitioned by your local Sperry Univac representative. To receive only the updating package, order UP-8748 Rev. 1-A. To receive the complete manual, order UP-8748 Rev. 1.

LIBRARY MEMO ONLY	LIBRARY MEMO AND ATTACHMENTS	THIS SHEET IS
Mailing Lists BZ, CZ and MZ	Mailing Lists A00, A07, A08, A09, B00, B07, B09, 18, 18U, 19, 19U, 20, 20U, 21, 21U, 28U, 29U, 75, 75U, 76, and 76U (Package A to UP-8748 Rev. 1, 22 pages plus Memo)	Library Memo for UP-8748 Rev. 1-A RELEASE DATE: September, 1982



PAGE STATUS SUMMARY

ISSUE: Update A – UP-8748 Rev. 1
RELEASE LEVEL: 8.0 Forward

Part/Section	Page Number	Update Level	Part/Section	Page Number	Update Level	Part/Section	Page Number	Update Level
Cover/Disclaimer		Orig.						
PSS	1	A						
Acknowledgment	1	Orig.						
Preface	1, 2	Orig.						
Contents	1 2, 3 4, 5	A Orig. A						
1	1 2 thru 4	Orig. A						
2	1 thru 10 11 12	Orig. A Orig.						
3	1 thru 11	Orig.						
4	1 thru 12	Orig.						
5	1 thru 14 14a 15 thru 19	Orig. Orig. Orig.						
6	1 thru 5	Orig.						
7	1 thru 23 24	Orig. A						
Appendix A	1, 2	A*						
Index	1 2 3 4	Orig. A Orig. A						
User Comment Sheet								

*New pages

All the technical changes are denoted by an arrow (→) in the margin. A downward pointing arrow (↓) next to a line indicates that technical changes begin at this line and continue until an upward pointing arrow (↑) is found. A horizontal arrow (→) pointing to a line indicates a technical change in only that line. A horizontal arrow located between two consecutive lines indicates technical changes in both lines or deletions.



Contents

PAGE STATUS SUMMARY

ACKNOWLEDGMENT

PREFACE

CONTENTS

1. INTRODUCTION

1.1.	OVERVIEW OF IMS AND DMS	1-1
1.1.1.	Types of Files and Data Structures IMS and DMS Can Access	1-1
1.1.2.	Languages Used to Write IMS Action Programs and DMS Application Programs	1-2
1.2.	THE IMS/DMS INTERFACE	1-2
1.3.	CAPABILITIES	1-3
1.4.	HOW IMS AND DMS INTERFACE	1-3
1.5.	SYMBOLS, NOTATIONS, AND DOCUMENTATION REFERENCES IN THIS MANUAL	1-4

2. IMS AND DMS PREPARATION, START-UP, AND SHUTDOWN

2.1.	GENERAL	2-1
2.2.	DMS SYSTEM GENERATION	2-1
2.2.1.	Device Media Control Language	2-2
2.2.1.1.	The DMCL IMS QUICK-BEFORE-LOOKS Sentence	2-3
2.3.	IMS CONFIGURATION	2-4

2.4.	DBMS START-UP PROCESSING	2-5
2.4.1.	MAXIMUM IMS-THREADS Statement	2-7
2.4.2.	MAXIMUM IMS-TERMINALS Statement	2-7
2.4.3.	QUICK-BEFORE-LOOKS Statement	2-8
2.4.4.	IMS QUICK-BEFORE-LOOKS Statement	2-9
2.5.	NORMAL IMS/DMS TERMINATION	2-11
2.6.	ABNORMAL IMS TERMINATION	2-11
2.7.	ABNORMAL DMS TERMINATION	2-12
3.	TRANSACTION PROCESSING	
3.1.	TRANSACTIONS IN AN IMS/DMS INTERFACE	3-1
3.2.	IMS SUCCESSION IN AN INTERFACE TRANSACTION	3-2
3.3.	IMS/DMS DIALOG TRANSACTIONS	3-3
3.3.1.	Rule 1 - Three Data Manipulation Language Verbs Are Prohibited	3-3
3.3.2.	Rule 2 - One IMS Success Unit Contains One DMS Updating Run Unit	3-3
3.3.2.1.	IMS Single-Thread Success Unit	3-4
3.3.2.2.	IMS Multithread Success Unit	3-5
3.3.3.	Rule 3 - Dialog Transactions Must Use BIND/UNBIND Verbs	3-6
3.3.3.1.	IMS Single-Thread Considerations for UNBIND and BIND Verbs	3-7
3.3.3.2.	IMS Multithread Considerations for UNBIND and BIND Verbs	3-7
3.4.	IMS CONTROL OVER DATA BASE ACCESS	3-9
3.5.	TRANSACTION LOCKS AND ROLLBACK	3-9
4.	WRITING DML/COBOL ACTION PROGRAMS	
4.1.	ACTION PROGRAM PREPARATION	4-1
4.1.1.	Preprocessing the Action Program	4-1
4.1.2.	Compiling the Action Program	4-3
4.1.3.	Linking the Action Program	4-3
4.1.4.	Job Control Stream to Preprocess, Compile, and Link an Action Program	4-4
4.2.	DATA MANIPULATION LANGUAGE IN A COBOL ACTION PROGRAM	4-5
4.2.1.	Data Division	4-5
4.2.1.1.	INVOKE Statement	4-6
4.2.1.2.	DMCA Clause	4-6
4.2.1.3.	COPYING RECORDS Clause	4-8
4.2.1.4.	Examples of DMCA and COPYING RECORDS Clauses	4-9
4.2.2.	Procedure Division	4-10

5. DATA BASE ACCESS VIA DEFINED RECORD MANAGEMENT

5.1.	CONSIDERATION IN USING DEFINED RECORD MANAGEMENT	5-1
5.2.	DATA DEFINITION DESCRIPTION	5-1
5.3.	PREPROCESSING A DATA DEFINITION	5-2
5.4.	DATA DEFINITION LANGUAGE	5-3
5.4.1.	Data Division	5-3
5.4.2.	Definition Division	5-4
5.5.	DEFINING A PARENT RECORD	5-5
5.5.1.	FROM DMS Statement	5-5
5.5.2.	VIA Statement	5-6
5.5.3.	OWNED BY Statement	5-7
5.5.4.	Optional TYPE IS Statement	5-8
5.5.5.	Optional PREFIX IS Statement	5-9
5.5.6.	ALLOW Statement	5-9
5.5.6.1.	ALLOW ADD Statement	5-9
5.5.6.2.	DELETE Clause	5-10
5.5.7.	Set Inclusion Definition	5-11
5.5.7.1.	WITHIN Statement for the Set Inclusion Definition	5-12
5.5.7.2.	OWNED BY Statement for the Set Inclusion Definition	5-13
5.5.7.3.	POINTER IS Statement for the Set Inclusion Definition	5-13
5.5.8.	Item Definition	5-14
5.5.9.	Supplement Definition	5-14a
5.5.9.1.	Supplement Definition for an Owner Record	5-15
5.5.9.2.	Supplement Definition for a Separate Data Base Record	5-16
5.5.9.3.	Supplement to a Record Outside the Data Base	5-17
5.6.	DEFINING A CHILD RECORD	5-18
5.6.1.	PARENT IS Statement	5-18
5.6.2.	FOLLOWS Statement	5-19

6. FILE RECOVERY

6.1.	RECOVERY FUNCTIONS	6-1
6.2.	ONLINE RECOVERY	6-2
6.2.1.	IMS-QBL File for Online Recovery	6-2
6.2.1.1.	IMS-QBL File Allocation	6-3
6.2.1.2.	IMS-QBL File Management	6-3
6.2.1.3.	IMS-QBL Recovery Procedure on Automatic Backward Recovery Failure	6-4
6.2.1.4.	IMS-QBL Quick Recovery	6-5
6.3.	OFFLINE RECOVERY	6-5

7. IMS/DMS EXAMPLE APPLICATION

7.1.	DEFINING THE EXAMPLE APPLICATION	7-1
7.2.	STRUCTURE OF THE EXAMPLE DATA BASE	7-1
7.3.	DMS SYSTEM PREPARATION	7-2
7.3.1.	Building the Data Dictionary	7-2
7.3.1.1.	Step 1: Generate DMCL for the Data Dictionary	7-3
7.3.1.2.	Step 2: Allocate and Initialize the Data Dictionary	7-4
7.3.1.3.	Step 3: Start Up DBMS for the Language Processors	7-5
7.3.1.4.	Step 4: Generate Schema for the Data Base	7-6
7.3.1.5.	Step 5: Generate DMCL for the Data Base	7-7
7.3.1.6.	Step 6: Generate Subschema for the Example Application	7-9
7.3.1.7.	Step 7: Shut Down DBMS	7-10
7.3.2.	Creating and Loading the User Data Base	7-10
7.3.2.1.	Step 1: Allocate and Initialize the Data Base	7-10
7.3.2.2.	Step 2: Restart DBMS	7-11
7.3.2.3.	Step 3: Compile COBOL/DML Program to Load the Data Base	7-12
7.3.2.4.	Step 4: Execute Program to Load the Data Base	7-15
7.4.	IMS SYSTEM PREPARATION	7-19
7.4.1.	Configuring IMS	7-19
7.4.2.	Compiling the Action Program	7-20
7.5.	EXECUTING THE EXAMPLE ACTION PROGRAM	7-23
7.6.	SHUTTING DOWN IMS AND DMS	7-24

APPENDIX**A. DMS ROLLBACK ERROR CODES RETURNED TO IMS****INDEX****USER COMMENT SHEET****FIGURES**

3-1.	IMS Single-Thread Success Unit	3-4
3-2.	DMS Run Unit Spanning IMS/DMS Success Units	3-5
3-3.	Dialog Transaction Using BIND/UNBIND Verbs	3-6
3-4.	Action Programs Multiple Paths Succession	3-8
4-1.	Sequence for Preprocessing and Compiling Action Programs	4-2

5-1.	Creating a Data Definition Record	5-2
5-2.	Source Record Relationship for a Defined Parent Record	5-8
5-3.	Example of Addition of Defined Record to Other Sets	5-10
5-4.	Example of Addition of Defined Record Using Key Value	5-12
5-5.	Owner Record Supplement Relationship	5-15
5-6.	Separate Data Base Record Supplement Definition	5-16
5-7.	Supplement to a Record Outside the Data Base	5-17
5-8.	Source Record Relationship for a Defined Child Record	5-19
7-1.	Data Structure Diagram	7-2
7-2.	Generating the DMCL for the Data Dictionary	7-3
7-3.	Allocating the Data Dictionary	7-4
7-4.	Initializing the Data Dictionary	7-4
7-5.	Starting Up the DBMS to Execute Language Processors	7-5
7-6.	Generating the Schema	7-6
7-7.	Generating the DMCL for the User Data Base	7-8
7-8.	Generating the Subschema	7-9
7-9.	Allocating the User Data Base	7-10
7-10.	Initializing the User Data Base	7-11
7-11.	Starting Up the DBMS for the Application	7-11
7-12.	Compiling COBOL/DML Program to Load the Data Base	7-12
7-13.	Loading the Data Base	7-15
7-14.	Job Control Stream to Configure IMS	7-20
7-15.	Input to the IMS Configurator	7-20
7-16.	Compiling the Action Program	7-21
7-17.	Starting Up IMS to Execute the Action Program	7-24

TABLES

3-1.	Termination/Lock and Rollback Status at Action Program Termination	3-10
------	--	------



1. Introduction

1.1. OVERVIEW OF IMS AND DMS

Two systems that provide for accessing and updating data are IMS and DMS. IMS provides online inquiry with updating capability; DMS provides access to and updating control of data bases stored on direct access devices. Both systems offer the data integrity and data security features needed in real time processing. These include:

- Logging of data modifications
- Online rollback of erroneous data modifications
- Security locks on data being updated
- Offline restoration of data files

1.1.1. Types of Files and Data Structures IMS and DMS Can Access

IMS action programs can access the following file structures:

- Sequential access method (SAM)
- Direct access method (DAM)
- Indexed sequential access method (ISAM)
- Multi-indexed random access method (MIRAM)
- Defined files (logical structure) via defined record management

The IMS uniform inquiry update element (UNIQUE) accesses files through defined record management only.

DMS data base records are stored in one or more system access technique (SAT) files. The permitted logical data base structures are:

- Sequential (list) structures
- Hierarchical (tree) structures
- Network structures

A data record can be a member of more than one structure simultaneously. This allows you to access data for a wide variety of applications. The description of the logical structure and access methods for an entire data base is called a schema and is written in the schema data description language (DDL). The subset of the logical data base referenced by any given application program or group of programs is described in a subschema by using the subschema DDL. Subschemas let you select only the data and structures you need for an application and provide a measure of data base security.

1.1.2. Languages Used to Write IMS Action Programs and DMS Application Programs

IMS action programs are written in COBOL, RPG II, or basic assembler language (BAL). IMS also offers UNIQUE, a comprehensive inquiry/update facility that requires no programming effort except to define the data structure.

DMS application programs are written in COBOL and the DMS data manipulation language (DML). These programs do not contain any data description for the data base records; instead, they specify the applicable subschema.

In IMS and DMS, I/O is handled by the respective data management system.

1.2. THE IMS/DMS INTERFACE

The IMS/DMS interface allows you to access a DMS data base from IMS action programs and UNIQUE. This gives the IMS user the advantages of DMS structural flexibility and powerful access mechanisms and gives the DMS user easy online access to the data base.

The IMS system you configure is either single-thread or multithread. In single-thread IMS systems, only one action is processed at a time, and actions for different transactions are interspersed. Since the duration of an action is normally short, IMS can handle transactions originating from several terminals concurrently, with very little increase in response time. Multithread IMS allows concurrent processing of actions for different transactions, with increased throughput.

The DMS system is multithread. When you combine it with the IMS multithread system, you can concurrently process actions for different transactions accessing the data base. → The DMS system can also handle more than one IMS system concurrently.

You can expect maximum performance when you use the combination of multithread IMS and DMS; however, single-thread IMS and multithread DMS provide acceptable performance in many instances.

1.3. CAPABILITIES

The IMS/DMS interface allows you to:

- access a DMS data base directly from your COBOL action programs; or
- use DMS files to build an IMS defined file accessible from your action programs and from UNIQUE.

The same action program can directly access DMS data bases and conventional data files. These file types are:

- ISAM
- MIRAM
- DAM
- SAM

An action program accessing a data base can also access an IMS defined file through defined record management, provided the defined file does not access the data base. That is, both types of data base access are not permitted within the same transaction. UNIQUE accesses data bases only through defined record management.

An IMS defined file can include all or parts of a DMS data base, as well as conventional ISAM, DAM, or MIRAM files.

The IMS/DMS interface has facilities for online and offline recovery.

1.4. HOW IMS AND DMS INTERFACE

DMS interfaces with IMS through the data manipulation language (DML) and the DML preprocessor. DML statements are embedded in COBOL action programs and IMS data definitions to permit access to the data base. Action programs and data definitions accessing a data base must be preprocessed by the DML preprocessor before you submit them to the COBOL compiler or the data definition processor.

Action programs accessing DMS through IMS defined record management do not require preprocessing. Defined files are accessed by CALL statements in either a COBOL or BAL action program. RPG II action programs use standard operations to access defined files.

An action program communicates with DMS through data stored in the data management communications area (DMCA) of the action program. Certain functions require that you store data in this area before calling DMS. After a call to DMS, the DMCA contains data concerning the outcome of the requested service. Certain error conditions produce different error codes depending on whether the call to DMS came from single-thread or multithread IMS; these codes are listed in Appendix A.

For more complete information on the DMCA, refer to the DMS data manipulation language user guide/programmer reference, UP-8036 (current version).

1.5. SYMBOLS, NOTATIONS, AND DOCUMENTATION REFERENCES IN THIS MANUAL

This manual does not attempt to repeat or replace any material that appears in the IMS or DMS manuals. Some material from those manuals overlaps the material in this manual because this manual draws the two systems together. Instead of repeating material, we reference the manual that provides the most detailed description.

The symbols and notations used in this manual are consistent with the symbols and notations used in the IMS and DMS manuals. The following format conventions are used in this manual:

1. Words that appear in all capitals are reserved words.
2. Underlined capitalized words are keywords and are required when the functions in which they appear are used, except in the case of defaults. Those capitalized words not underlined are optional, and you may include them to improve readability. All capitalized words are part of the languages and must be spelled exactly as indicated.
3. Capitalized words with a double underline begin a statement and must start on a new line.
4. Lowercase words are generic terms you must supply.
5. Braces { } indicate that you must choose one of the elements within the braces.
6. Optional functions are enclosed in brackets [].
7. Periods must be used where indicated.
8. An ellipsis ... indicates optional repetition of the preceding syntax element.

3. When you omit the ON ERROR clause, the option specified in the device media control language source is used.

2.5. NORMAL IMS/DMS TERMINATION

Usually, you shut down IMS before shutting down DMS. If you try to shut down DMS while accesses to the data base are still taking place, the shutdown is not performed and you receive an error message indicating the data base is still in use.

In some circumstances, however, you can shut down DMS without first terminating the IMS session. This is because DMS treats IMS as a special case. DMS honors your shutdown request provided:

- All DMS batch applications have terminated.
- IMS is not accessing the data base.

This allows you to shut down DMS in an orderly fashion while still letting IMS continue with its non-DMS work.

The master terminal operator shuts down IMS by entering either of two commands:

ZZSHD (for normal termination)

or

ZZHLT (for emergency only)

See the IMS system support functions user guide/programmer reference, UP-8364 (current version).

The operator shuts down DMS by entering this unsolicited command at the console:

```
n0 SHUTDOWN DBMS.
```

The variable n in this command is the job slot number of the DBMS job and 0 indicates to OS/3 that this is an unsolicited command. See the DMS system support functions user guide/programmer reference, UP-8272 (current version).

2.6. ABNORMAL IMS TERMINATION

When IMS is shut down (by ZZHLT, ZZSHD, or an error causing abnormal termination), the OS/3 operating system notifies DMS that IMS has terminated (whether IMS termination is normal or abnormal). DMS initiates processing of active run-unit termination and rollback, but DMS remains running. To reestablish the interface, restart IMS by using the warm or cold restart procedure. For a detailed description on restarting IMS, refer to the IMS system support functions user guide/programmer reference, UP-8364 (current version).

2.7. ABNORMAL DMS TERMINATION

When DMS abnormally terminates, the OS/3 operating system notifies IMS that an abnormal DMS termination occurred. IMS continues running while DMS shuts down. IMS cancels any active transactions that access the data base, rolls back updates, and sends an error message to the terminals initiating those transactions. Transactions attempting to access the data base after the DMS shutdown are also canceled.

You must restart DMS before you can access the data base using IMS action programs. For a detailed description on restarting DMS, refer to the DMS system support functions user guide/programmer reference, UP-8272 (current version).

7.5. EXECUTING THE EXAMPLE ACTION PROGRAM

With the DBMS still loaded, the job control stream in Figure 7-17 starts up IMS in offline batch mode to execute the example action program. Fifty input messages are embedded in the job control stream. The transaction code STATE in each message initiates the action program to retrieve the desired data from the data base.

In an online IMS environment, the input messages would be entered interactively at IMS terminals and you would load ICAM before starting up IMS. Refer to the IMS system support functions user guide/programmer reference, UP-8364 (current version) for the complete start-up procedure.

```
// JOB IMSMT,,16000,,4
// DVC 20    // LFD PRNTR
// DVC 21    // LFD PRNTR1
// DVC 50    // VOL IMSDMS    // LBL *DBA LIBRARY*    // LFD DBALIB
// DVC 50    // VOL IMSDMS    // LBL AUDFILE    // LFD AUDFILE
// DVC 50    // VOL IMSDMS    // LBL CONDATA    // LFD CONDATA
// DVC 50    // VOL IMSDMS    // LBL NAMEREC    // LFD NAMEREC
// EXEC IMSMT,DBALIB
// PARAM BATCH=OFFLINE
/8
STATE ALABAMA
STATE ALASKA
STATE ARIZONA
STATE ARKANSAS
STATE CALIFORNIA
STATE COLORADO
STATE CONNECTICUT
STATE DELAWARE
STATE FLORIDA
STATE GEORGIA
STATE HAWAII
STATE IDAHO
STATE ILLINOIS
STATE INDIANA
STATE IOWA
STATE KANSAS
STATE KENTUCKY
STATE LOUISIANA
STATE MAINE
STATE MARYLAND
STATE MASSACHUSETTS
STATE MICHIGAN
STATE MINNESOTA
STATE MISSISSIPPI
STATE MISSOURI
STATE MONTANA
STATE NEBRASKA
STATE NEVADA
```

Figure 7-17. Starting Up IMS to Execute the Action Program (Part 1 of 2)

```
STATE NEW HAMPSHIRE
STATE NEW JERSEY
STATE NEW MEXICO
STATE NEW YORK
STATE NORTH CAROLINA
STATE NORTH DAKOTA
STATE OHIO
STATE OKLAHOMA
STATE OREGON
STATE PENNSYLVANIA
STATE RHODE ISLAND
STATE SOUTH CAROLINA
STATE SOUTH DAKOTA
STATE TENNESSEE
STATE TEXAS
STATE UTAH
STATE VERMONT
STATE VIRGINIA
STATE WASHINGTON
STATE WEST VIRGINIA
STATE WISCONSIN
STATE WYOMING
/*
/b
// FIN
```

Figure 7—17. Starting Up IMS to Execute the Action Program (Part 2 of 2)

7.6. SHUTTING DOWN IMS AND DMS

Because IMS was started up in offline batch mode, you do not have to shut down IMS. In an online environment, you would shut down IMS with the master terminal command:

```
ZZSHD
```

The final step is to shut down the DBMS with the operator command:

```
→ nØ SHUTDOWN DBMS.
```

Appendix A. DMS Rollback Error Codes Returned to IMS

When an error occurs during the execution of a DMS verb, DMS informs the calling program by returning an error code in the DMCA, in either or both of the fields ERROR-STATUS and RB-ERROR-CODE. If either of these fields contains a non-zero value, an error has occurred.

DMS errors are either fatal or nonfatal. For fatal errors, both ERROR-STATUS and RB-ERROR-CODE contain nonzero values. For nonfatal errors, only ERROR-STATUS contains a nonzero value.

DMS run units automatically check for fatal errors because the DML preprocessor inserts an "IF RB-ERROR-CODE NOT EQUAL TO ZEROS GO TO . . ." statement after each DMS verb. This statement directs program control to the rollback paragraph that the user specified in the INVOKE statement. To check for nonfatal errors, the user should include a "PERFORM DMS-STATUS." statement after each statement in his program that contains a DML verb.

When a nonfatal error occurs (that is, RB-ERROR-CODE is zero but ERROR-STATUS is not), the first two bytes of ERROR-STATUS contain the major verb code and the last two bytes indicate the specific error. For the interpretation of these codes, see the data base management system (DMS) data manipulation language user guide/programmer reference, UP-8036 (current version).

When a fatal error occurs, the last two bytes of ERROR-STATUS contain "59", signifying that DMS has forced a DEPART WITH ROLLBACK on behalf of the run unit. The RB-ERROR-CODE field, which is also divided into two parts, provides further information. The last two bytes of RB-ERROR-CODE give the reason for the rollback, and the first two show whether the rollback succeeded. If these first two bytes contain zeros, the rollback succeeded; otherwise, the rollback failed for the reason indicated. The codes returned in RB-ERROR-CODE are described in the DMS data manipulation language user guide/programmer reference, UP-8036 (current version).

↓

For multithread IMS (as well as for batch DMS run units), the use of the DMCA error fields is as described in this appendix and in the current version of UP-8036. Single-thread IMS generally follows the same conventions, but there are exceptions. For a few fatal error conditions, IMS returns a slightly different error code. If the last two bytes of ERROR-STATUS are "59", you should interpret the RB-ERROR-CODE field as previously described. If, however, the last two bytes of ERROR-STATUS are zero, the first two bytes of RB-ERROR-CODE have the following meaning:

<u>Last Two Bytes of RB-ERROR-CODE</u>	<u>Normal DMS Equivalent</u>	<u>Meaning</u>
01	59	IMS not allowed with DMS
02	97	Unable to call DMS
03	40	IMS terminal limit exceeded
04	99	Run unit not bound

NOTE:

Do not confuse the "59" shown in this list with the "59" that may appear in the last two bytes of ERROR-STATUS. These are two unrelated uses of the number "59".

For these four error conditions, IMS also sends the terminal operator a message describing the error. These messages are:

<u>DMCA Error Code</u>	<u>IMS Message</u>
0001	DMS02 DMS WOULD NOT PERMIT IMS ACCESS
0002	DMS01 IMS COULD NOT CALL DMS SYMBIONT
0003	DMS05 IMPART WAS NOT SUCCESSFUL
0004	DMS09 ACTION DID NOT ISSUE IMPART FIRST

↑

Index

Term	Reference	Page	Term	Reference	Page
A			B		
Abnormal termination			BIND statement	3.3.3	3-6
lock-rollback indicator	3.5	3-9			
online recovery	6.2	6-2			
transaction rollback	3.5	3-9			
ACCESS-CONTROL			C		
KEY	2.4.4	2-10	CALC		
LOCK	2.4.3	2-8	FROM DMS statement	5.5.1	5-5
Action program, IMS			OWNED BY statement	5.5.3	5-7
compiling	4.1.2	4-3	set inclusion definition	5.5.7.1	5-12
example	7.4.2	7-20	supplement definition	5.5.9.2	5-16
executing	Fig. 7-16	7-21	Child record definition	5.6	5-18
job control stream	7.5	7-23	CLOSE statement	4.2.2	4-12
languages	4.1.4	4-4	COBOL action program		
linking	1.1.2	1-2	compiling	4.1.2	4-3
multiple path succession	4.1.3	4-3	linking	4.1.3	4-3
preprocessing	3.3.3.2	3-8	serially reusable	4.2.1.2	4-7
programming rules	4.1.1	4-1	sharable	4.2.1.2	4-7
ALLOW statement	3.3.1	3-3	Commit point	3.1	3-1
	5.5	5-5	Concurrent processing	1.2	1-2
	5.5.6	5-9	Configuration		
Application programs, DMS	1.1.2	1-2	considerations	2.3	2-4
Audit-continuity data file (AUDCONF)	6.1	6-1	example	Fig. 7-14	7-20
Automatic backward recovery			example	Fig. 7-15	7-20
failure	6.2.1.3	6-4	Conventional data files	1.3	1-3
online	2.2.1.1	2-3	COPYING RECORDS clause		
Automatic set relationship	5.5.7.1	5-12	example	4.2.1.4	4-9
			format	4.2.1.3	4-8

Term	Reference	Page	Term	Reference	Page
D					
Data base			Dialog transaction		
accessed from action program	4.1	4-1	description	3.3	3-3
allocating and initializing	7.3.2.1	7-10	using BIND/UNBIND example	Fig. 3-3	3-6
defined record management access	5.1	5-1	DMCA		
loading	7.3.2.4	7-15	action program preparation	4.2.1.2	4-7
Data base management system (DBMS)			data definition	4.2.1.2	4-7
example start-up	Fig. 7-5	7-5	example	4.2.1.4	4-7
	Fig. 7-11	7-11	purpose	1.4	1-3
shutdown	2.5	2-11	DMCL		
start-up	2.4	2-5	data dictionary	7.3.1.1	7-3
Data definition language			user data base	7.3.1.5	7-7
data division format	5.3	5-2	DML preprocessor		
defined record definition	5.2	5-1	data definition	5.3	5-2
Data definition preprocessing	5.3	5-2	IMS action program	4.1.1	4-2
Data definition processor	5.3	5-2	DMS-ABORT section	4.2.2	4-11
Data dictionary	7.3.1	7-2	DMS error codes	Appendix A	
Data division, COBOL action program	4.2.1	4-5	DMS-STATUS FILE clause	4.1.1	4-3
Data manipulation language			DMS-SUCCESS section	4.2.2	4-11
IMS action programs	4.1	4-1			
intercepted by IMS	5.3	5-2	E		
preprocessor	1.4	1-3	Error processing	Appendix A	
Data structures			Example application	Section 7	7-1
DMS data base	1.1.1	1-2	External succession	3.2	3-2
example data base	7.2	7-1			
IMS defined file	5.2	5-1	F		
Defined file			File access methods	1.1.1	1-1
accessed by action programs	5.1	5-1	File recovery	6.1	6-1
derivation from subschema	5.5.1	5-6	FOLLOWS statement	5.6.2	5-19
structure	4.2	4-5	FROM DMS statement	5.5.1	5-5
Defined record definition	5.2	5-1			
Defined record management					
data base access	5.2	5-1			
file access	1.1.1	1-1			
Delayed internal succession	3.2	3-2			
DELETE clause	5.5.6.2	5-10			
DEPART statement					
establishing rollback point	3.3.2	3-3			
intercepted by IMS	3.3.2.2	3-5			
programming rules	3.3.2	3-3			
Device media control language (DMCL)	2.2.1	2-2			

Term	Reference	Page	Term	Reference	Page
I			M		
IMPART statement			MANUAL set relationship	5.5.7.1	5—12
intercepted by IMS	3.3.2.2	3—5	MAXIMUM IMS-TERMINALS statement	2.4.2	2—7
programming rules	3.3.2	3—4	MAXIMUM IMS-THREADS statement	2.4.1	2—7
IMS configuration considerations	2.3	2—4	Multithread		
	7.4.1	7—19	IMS/DMS	1.2	1—2
IMS system preparation	7.4	7—19	run units and success units	3.3.2	3—4
IMS QUICK-BEFORE-LOOKS statement					
format	2.2.1.1	2—3	O		
online	6.2.1	6—2	Offline recovery	6.3	6—5
Internal succession	3.2	3—2	ON ERROR clause	2.4.3	2—8
INVOKE statement			Online recovery	6.2	6—2
action program	4.2.1.1	4—6	OWNED BY statement		
data definition	5.4.1.1	5—4	defined record definition	5.5.3	5—7
Item definition	5.5.8	5—14	set inclusion definition	5.5.7.2	5—13
			Owner record supplement definition	5.5.9.1	5—15
J			P		
Job control stream (action programs)	4.1.4	4—4	PARENT IS statement	5.6.1	5—18
Journal file, offline recovery	6.1	6—1	Parent record	5.5	5—5
Journaling	6.1	6—1	POINTER statement	5.5.7.3	5—13
			PREFIX IS statement	5.5.5	5—9
L			PROCEED mode, ON ERROR clause	2.4.3	2—9
Lock-rollback-indicator, online recovery	3.5	3—9			

Term	Reference	Page	Term	Reference	Page
Q			T		
Quick-before-look file (QBL) format	2.4.3	2—8	Terminal output message file (TOMFILE)	6.1	6—1
online recovery	6.2.1	6—2	Termination abnormal	2.6	2—11
R			normal	2.5	2—11
Recovery	6.1	6—1	Termination/lock and rollback status	Table 3—1	3—10
ROLE IN UPDATE statement	5.5.9.1	5—16	Transaction description	3.1	3—1
Rollback			locks and rollback	3.5	3—9
codes	Appendix A		TYPE IS statement	5.5.4	5—8
online recovery	3.5	3—9	U		
transaction rollback	3.1	3—1	UNBIND statement	3.3.3	3—6
Rules for dialog transactions	3.3	3—3	UNIQUE	5.1	5—1
Run unit	3.3.2	3—3	V		
S			VIA statement	5.5.2	5—6
Schema, generating	7.3.1.4	7—6	W		
Schema section, DML preprocessor	5.3	5—2	WITHIN statement	5.5.7.1	5—12
Set inclusion definition	5.5.7	5—11			
Set relationships	5.5.7.1	5—12			
Shutdown	2.5	2—11			
	7.6	7—24			
Single-thread processing	1.2	1—2			
Start-up, IMS/DMS	2.1	2—1			
STOP mode, ON ERROR clause	2.4.3	2—8			
Subschema					
data definition source	5.2	5—1			
generating	7.3.1.6	7—9			
invoked by action program	4.2.1.1	4—6			
Success unit					
multithread	3.3.2.2	3—5			
single thread example	Fig. 3—1	3—4			
Succession	3.2	3—2			
Supplement definition	5.5.9	5—14			
Synchronized success unit	3.1	3—1			
System access technique file (SAT)	1.1.1	1—2			

USER COMMENT SHEET

Your comments concerning this document will be welcomed by Sperry Univac for use in improving subsequent editions.

Please note: This form is not intended to be used as an order blank.

(Document Title)

(Document No.)

(Revision No.)

(Update No.)

Comments:

ong line.

From:

(Name of User)

(Business Address)

Fold on dotted lines, and mail. (No postage stamp is necessary if mailed in the U.S.A.)
Thank you for your cooperation

FOLD



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 21 BLUE BELL, PA.

POSTAGE WILL BE PAID BY ADDRESSEE

SPERRY UNIVAC

ATTN.: SYSTEMS PUBLICATIONS

P.O. BOX 500
BLUE BELL, PENNSYLVANIA 19424



CUT

FOLD