

The Plain-English Definitive Guide to Quickly Resolving CA-IDMS[®] Program Problems

Release 5 - March 11, 1999

**Copyright 1999 by Neal Walters, Amerisoft Inc.
This document may be freely copied and distributed
provided no changes are made and all pages are included.**

Free Download of this book available at web site below.

**To Learn more about IDMS,
try "It-Does-More-Stuff" tutorials. V158**

**<http://www.ItDoesMoreStuff.com>
The web site dedicated to IDMS.**

**Free Discussion Forums!
Free Utilities!
And More!
see page 49 for details**

Please send suggestions for new error
descriptions and updates to
errorbook@itdoesmorestuff.com

or to request further information
nwalters@itdoesmorestuff.com
888-240-4515 or 972-771-4091

Purpose: This document attempts to be a easy-to-use reference guide to help quickly solve IDMS problems. The 80/20 rules applies to IDMS - in that 80% of the problems can be answered in 20% of the error books.

Many companies no longer provide a complete set of manuals to their programmers, or the manuals are now online on CD-ROM or “IBM reader” on TSO. This document is an attempt to solve 80% or more of the problems with one quick reference guide.

Another feature of this book is 17 years of experience. Many of the IDMS manuals give you a “bland” definition of the error - but don’t tell you what the possible causes or fixes are. This book attempts to tell you an extra two or three important things about each error message.

If this document can get into the hands of 1000 people, and each person saves 8 hours per year by using this document, then that is a total of 8000 hours. For sake of illustration, let’s say the average IDMS professional is billed at \$50/hour. Thus this document will make the world economy \$400,000 richer!

Call for HELP: Rather than this book just being my 17 years experience, I would like to make it hundreds of years of experience from hundreds of people. If you have a common/repetitive error situation that is not covered in this book, or if you have a unique description or solution of an IDMS problem, PLEASE email me at errorbook@itdoesmorestuff.com. This book will be regularly posted to the <http://www.ItDoesMoreStuff.com> web site, available for download in Adobe Acrobat .PDF format.

Legal Notices: IDMS and CA/IDMS are registered trademarks and products of Computer Associates International, Inc. All other names and products are trademarks or registered trademarks of their respective holders. Author is not responsible for any problems caused by using or misusing advice in this booklet.

Table of Contents

TABLE OF CONTENTS	3
TERMINOLOGY	5
CV	5
IDMS JOURNALS.....	5
LOGS	5
AREAS (UPDATE/RETRIEVAL/OFFLINE).....	5
VTAM APPLID	5
STARTUP/SHUTDOWN	6
LOCAL MODE UPDATE JOBS.....	6
BATCH/CV SHARED	6
BATCH/CV PROTECTED.....	6
DCMT	6
PMRM OR OPER.....	6
UCF/BATCH	6
SELECT LOCKS	6
UPDATE LOCKS	6
DEADLOCK.....	7
ROLLBACK.....	7
DML	7
DML PRECOMPILER (OR PREPROCESSOR).....	7
STAGING AND MIGRATION	7
CICS.....	7
ABEND	7
IDMS ABENDS.....	8
SELECTED ABND#### (4 DIGIT) ABEND MESSAGES	8
SELECTED DC##### (6 DIGIT) ABEND MESSAGES	10
<i>DC001003 - STALLED WAITING FOR</i>	11
<i>DC027007 - TASK ABENDED</i>	11
<i>DC201006</i>	11
<i>DC027002</i>	12
<i>DC001000</i>	12
<i>DC001002</i>	12
DC001003 - CASE STUDY #1 (STALLED WAITING).....	13
DC001000 - CASE STUDY #2 (DEADLOCK)	15
DC027007 - CASE STUDY #3 (ABRU).....	16
IDMS ERROR-STATUS.....	17
GENERAL FORMAT	17
EACH MAJOR-CODE REPRESENTS AN IDMS DML VERB	17
IDENTIFYING IDMS ERRORS IN JOB OUTPUT (SYSOUT).....	18
ADDITIONAL IDMS DIAGNOSTIC INFORMATION IN JOB OUTPUT.....	19
COBOL CODE: FINDING DML-SEQUENCE AND ERROR HANDLING LOGIC	20
FREE ERROR HELPS UTILITY	21
HANDLING OF MOST COMMON ERROR STATUSES.....	22
COMMON ERRORS CORRECTABLE BY DATABASE ADMINISTRATION:.....	22
COMMON ERRORS CORRECTABLE BY OPERATIONS:	23
COMMON ERRORS CORRECTABLE BY PROGRAMMING:	24
DETAILED ERROR MESSAGES - BY MINOR CODE.....	25

IDMS ERROR SUBROUTINE	42
THE WHEN-COMPILED AND WHERE-COMPILED VARIABLES:	45
TYPES OF ERROR EXPLANATIONS:.....	46
REPORTING PROBLEMS TO CA AND USING TCC	48
ITDOESMORESTUFF WEB SITE	49
FREE UTILITIES	49
COMPUTER BASED TRAINING TUTORIALS FOR IDMS	49
IDMS DISCUSSION FORUMS	49
IDMS JOB SEARCH	49
INDEX.....	50

Terminology

CV

CV stands for Central Version, perhaps called a “server” in modern parlance. IDMS (Integrated Database Management System) is a database server that concurrently services database requests to many different application programs (CICS online or CV batch). Local jobs are said to run “outside of the CV.” A CV handles multiple updates from multiple online and batch jobs to the same database at the same time. CV are usually assigned number between 1 and 255 (but most often 01 through 99).

Fill out this table as a reference of the CVs in your environment:

CV##	Use (test, QA, prod, etc...)	CICS Jobname	IDMS Jobname	VTAM Applid

Figure 1 - Table of CVs

IDMS Journals

IDMS Journals contain a history record of all before and after images of all database records. There are four disk journals per each CV. These are used for a ROLLBACK when a CV update job ABENDs. As these disk journals fill up, archive journal jobs are submitted (Jobname(s)=CV##AJNL). It is critical that these jobs run and offload the journals before the disk journals fill-up. Many sites use an ASG product called DB/TRACER or a CA product called Journal Analyzer to create reports from these journal tapes.

Logs

IDMS Logs contain primarily system messages and ABEND dumps (and sometimes statistical information). There is only one disk log file, which is regularly offloaded. It is critical that these jobs run and offload the IDMS LOG file before the log file fills-up.

Areas (Update/Retrieval/Offline)

An area in Update Mode allows multiple programs to simultaneously update the area. When an area in Retrieval Mode or Offline, programs that attempt to access it in update mode will get an IDMS error-status 0966.

VTAM Applid

The code used to access **Error! Bookmark not defined.**the IDMS CV from a 3270 terminal session.

Startup/Shutdown

A The process of bringing a CV up or down is called often called STARTUP and SHUTDOWN.

Local Mode Update Jobs

Local mode update jobs MUST be preceded by a job to vary the areas to OFFline or REtrieval mode followed by a backup job. Local jobs can run when the IDMS CV is down. If a local job ABENDS, the IDMS files must be restored by a pre-built restore job. Only one local mode update job may be updating a single area at one time.

Batch/CV Shared

These are the normal update jobs that can run concurrently. The IDMS CV must be up and the areas must be in UPDATE mode.

Batch/CV Protected

Protected jobs lock the area and do not permit other jobs to run against the same area at the same time. The area must be in UPDATE mode.

DCMT

Stands for Data Communications Master Terminal? This is the command that shows the status of most everything within IDMS. Start with "DCMT HELP" and go from there. Example DCMT commands are shown on pages **Error! Bookmark not defined.,Error! Bookmark not defined.,Error! Bookmark not defined.,Error! Bookmark not defined.** The DCMT task code is usually followed by a "V" for VARY or a "D" for DISPLAY. An example command would be:- DCMT V AREA EMPDEMO.EMP-DEMO-REGION RET.

PMRM or OPER

PMRM is the task code for the PERF-MON - CA-IDMS PERFormance MONitor. OPER is another task that can be used for monitoring.

UCF/Batch

This refers to the type of jobs that run in batch that do DCMT VARY AREA commands to take areas OFFLINE or to Vary Areas back to Update.

Select Locks

IDMS holds select (inquiry) locks on each record that is retrieved in an IDMS area that the program has readied in UPDATE mode.

Update Locks

IDMS holds update (exclusive) locks on each record that is updated in an IDMS area that the program has readied in UPDATE mode. The purpose of locks is to prevent a second program from updating a record that the first program has just updated. The first program might ABEND and the updated record might have to be rolledback (reversed). These locks are released when the COMMIT command is done. Not doing enough COMMITs is a frequent cause of program and

system problems. The locks tables and memory can be full because of the a number of locks. Unfortunately, the use of COMMITs requires that the program be made “restartable”, which requires more programming.

Deadlock

A condition that occurs when program #1 has locked records needed by program #2 - and visa versa - and both programs cannot wait indefinitely.

Rollback

When a program ABENDs IDMS automatically “rolls-back” or reverses all database updates made by the program (since the last COMMIT).

DML

Data Manipulation Language - these are the “IDMS” commands inserted into COBOL programs that allow the program to navigate the IDMS database (retrieve data, walk sets, update and store records, etc...). Examples of DML verbs are: READY, BIND, OBTAIN, STORE, ERASE, MODIFY, CONNECT, ROLLBACK, FINISH. See list of DML verbs (and “major codes”) on page 17.

DML Precompiler (or Preprocessor)

All COBOL programs that use DML commands must be preprocessed by the DML Precompiler before going through the COBOL compiler. The precompiler turns an command like “OBTAIN FIRST WITHIN SET” to a “CALL ‘IDMS’ USING”.

Staging and Migration

New COBOL programs go from the test system to the staging system, then from the staging system to production. This process is controlled by CLISTs and the Production Control group. These CLISTs include JCL to run the CA Migrator to migrate IDD record and file layouts from one IDMS dictionary to another.

CICS

CICS is an online teleprocessing monitor design to process inquiry and update transactions. Our users log on to CICS and run CICS programs which in turn access IDMS databases. At Raytheon Systems, there is a one-to-one relationship between each CICS and each IDMS-CV. See the Figure 1 - Table of CVs on page 5 for a complete list of CICS jobname and their relationship to an IDMS CV.

ABEND

ABEND is a computer word for “Abnormal End” - referring to when a program dies or crashes due to errors that are often outside of the program’s control. The ABEND might be due to a coding problem or an operational problem.

VDB

A third-party tool from Allen Systems groups that allows a database to be shadowed with a smaller VSAM file.

IDMS ABENDS

ABND#### - See description of error in IDMS Error Codes and Messages

Selected ABND#### (4 digit) Abend Messages

These types of errors usually do not show up in the output of the ABENDING program. They are found by viewing the IDMS CV job log or the outputs of the archive log jobs. Usually the DBA has to get involved to fix these types of problems.

ABND	Description	Solutions
D002	A DC##### error was issued with a severity code of 3, 4, or 5 causing the IDMS system to terminate the task.	Look for the preceding message on the log. See sample on page: ____ You see this message a lot - but it doesn't really mean anything without the associated DC##### message.
D003	A program check has occurred during run-unit processing. Probable storage overwritten.	See the DBA. This message is very rare.
D004	Run-away task - has been terminated by the IDMS system because it has executed longer than the time specified in the SYSGEN "RUNAWAY INTERVAL" parameter without an intervening wait.	Possible cause is a loop in the application program. This usually applies to online IDMS tasks, not CICS or batch jobs.
D005	IDMS/DC program has invoked an invalid CA/IDMS function or specified an invalid address for a valid function.	Does not apply to CICS/BATCH programs.
D009	Task terminated by the IDMS system because it exceeded its register stack space.	1) Program probably did too many updates (held too many locks) between COMMITs. 2) See SYSGEN "STACKSIZE" parameter.

Figure 2 - IDMS System-Level ABENDs (Such as D####)

11## 30##	Usually indicates a corrupt database or a problem left behind by a DBA change.	Contact the DBA.
3005	An attempt to open a database file has failed. Check the JCL for a missing or invalid DD statement or a mismatch between the DMCL page size and the disk data set blocksize.	Usually accompanied by a "MISSING DD" message. Often the DDNAME is misspelled or mistyped.
3010	An attempt to read from the database or disk journal has failed: 1) a physical I/O error, 2) attempt to read a relative block outside of the range, 3) failed attempt to initialize the file, 4) with VSAM, insufficient space in the storage pool	Contact the DBA - usually a problem with the segment and/or DMCL definitions or physical data set allocations.
ABRU	An external request unit executing under SYSCTL has abended. The abend may have been caused when the requested unit exceeded the maximum time allowed between database requests, or it may have been caused by other factors, as described in the system log. If the abend condition was detected by the check-user, lookaround-time, or external-wait mechanism, the system writes a SNAP DUMP to the log file.	
CKUR	The CheckUser task has detected that the program ABENDED. The IDMS "Check User" tasks monitors a batch job that have started IDMS run-units, and if one of those jobs ABENDs for any reason, then IDMS terminates the related run unit.	Usually an MVS ABEND occurred in the related batch job (such as a SOC#, or an operator cancelled).

Figure 3 - Other IDMS System-Level ABENDS

Selected DC##### (6 digit) Abend Messages

<p>DC001003</p>	<p>Program ABENDEd because it exceeded a wait time on a resource, such as lock (a DBKEY, page lock, or area lock).</p> <p><u>EXAMPLE 1:</u> T40 TASK:RHDCNP3S PROG:GDMGT713 STALLED WAITING FOR LMGR 8000 C29C51</p> <p><u>EXAMPLE 2:</u> T101 TASK:RHDCNP3S PROG:*ERUS* STALLED WAITING FOR LMGR 0 93120B02</p> <p><u>EXAMPLE 3:</u> T101 TASK RHDCRUAL STALLED WAITING FOR EREECB AT 9982510</p>	<p>Check the message to see which type of resource it is: 0=dbkey, 2000=page lock, 8000=area lock.</p> <p>1) Since LMGR is followed by 8000, C29C51 is a page number not a dbkey. This now converts to 12,754,001. This is the low-page-number of the area on which this job died waiting. A lookup of the page numbers shows that it was waiting on the TABLE-AREA.</p> <p>2) Since LMGR is followed by 0, then the second number (93120B02) is a dbkey.</p> <p>3) If LMGR is followed by a 2000, then the second number is a page# and represents a page lock.</p> <p>Example 3 - shows how the program could die waiting on some other resource. This could happen if CICS was having problems.</p>
-----------------	--	--

Figure 4 - IDMS ABENDs - DC#####

DC001003 - STALLED WAITING FOR

DC001003 TASK: <task-code> PROG: <program-name> STALLED WAITING FOR
<resource>

Reason: The task has exceeded the inactive interval time, the internal wait time, or the external wait time. These time limits are specified in the sysgen by the INACTIVE INTERVAL, INTERNAL WAIT, and EXTERNAL WAIT parameters of the SYSTEM statement. The inactive interval for an individual task can also be specified in the sysgen by the INACTIVE INTERVAL parameter of the TASK statement. If this condition occurs repeatedly, increase the appropriate time limit. For further information on these parameters, refer to CA-IDMS System Generation.

If the resource is LMGR see message DC001000 for meaning of LMGR yyyy zzzzzzzz

yyyy=8000 means an area lock and zzzzzzzz is the first page in the area,
yyyy=2000 means a dbkey lock and zzzzzzzz is a dbkey,
yyyy=0000 means a page lock and zzzzzzzz is the page number.

DC027007 - TASK ABENDED

DC027007 TASK: <task-code> PROG: <program-name> ABENDED WITH
<abend-code>

Reason: The named task has abended. For more information, see the description of the indicated task or user abend code.

DC201006

DC201006 CV_STATUS PROGNAME SUBSCHEM --RU_TASK_ __LOCAL-IDENT-- PRI

Reason: This message establishes a title line and supplies variable data in response to a DISPLAY CENTRAL VERSION request. The DISPLAY CENTRAL VERSION request can be entered by the operator or it can be generated internally (for example, as the result of an abort). The DC201006 message identifies the title line and all subsequent data lines. One data line appears for each external request unit controlled by the CA-IDMS central version. Each data line displays the status of a given run unit, followed by statistics for that run unit. For a complete description of the variables, refer to the discussion of the DISPLAY CENTRAL VERSION command in CA-IDMS System Tasks and Operator Commands.

DC027002

DC027002 SYSTEM MODULE PROGRAM CHECKED NEAR <vector-number> AT OFFSET
<hex-offset> PSW WAS <program-status-word> TASK: <task-code> PROG:
<program-name> DUMP OF SYSTEM FOLLOWS

Reason: A system internal error has occurred, possibly caused by storage
having been overwritten. Contact Computer Associates Technical Support.

DC001000

DC001000 T:<task-id> <task-name> P:<program-name> C:<cond> WAITING ON
R:<resource-type> <resource-id>

Reason: The named task is waiting on the named resource and is involved in
a deadlock. The <cond> parameter specifies whether the control is returned
to the waiting task (C:DEAD), or whether the task is aborted (C:NONE). The
<resource-id> is the resource for which the task was waiting when the
deadlock was detected. Resources include:

- Storage-
- Subschemas
- Program pool pages
- Area locks
- Database-key locks

Most resource IDs are self-explanatory, however, the resource ID specified
for the LMGR LK resource type contains information which can be used to
tune a database. The resource ID for LMGR LK is defined as follows:

R:LMGR LK <xxxxyyyy zzzzzzzz>

where: <xxxx>, = page group of the lock being requested

<yyyy> = lock code which identifies the value
in <zzzzzzzz>

<zzzzzzzz> = db-key, <yyyy> is 0000
lowpage of area (area lock), <yyyy> = 8000
page nbr (page lock), <yyyy> = 2000

One DC001000 message is issued for each task involved in a deadlock;
followed by a DC001002 message for each task that is selected as a
deadlock victim in order to resolve the deadlock.

DC001002

DC001002 T:<task-id> <task-name> P:<program-name> C:<cond> DEADLOCKED
ON R:<resource-type> <resource-id>

This message is issued in conjunction with message DC001000
to identify the task which has been selected as the deadlock
victim. The run unit associated with this task is rolled out
and control is returned to the waiting task if C:DEAD, other-
wise the task is aborted.

DC001003 - Case Study #1 (Stalled Waiting)

JES Messages / Output of CV Job:

```
3.48.48 JOB08303 +IDMS DC001003 V02 T59 TASK:RHDCNP3S PROG:XXXGL240 STALLED WAITING FOR LMGR 8000 40D1C1
3.48.48 JOB08303 +IDMS DC027007 V02 T59 TASK:RHDCNP3S PROG:XXXGL240 ABENDED WITH CODE D002
3.48.48 JOB08303 +IDMS DC201006 V02 T1 CV-Status BE-TaskID Pri FE - ID1 FE - ID2 FE TaskCD FE UserID
3.48.48 JOB08303 +IDMS DC201006 V02 T1 ABRT D002 59 100 CICSBULK CICS AE02 L140 ABC123
3.48.48 JOB08303 +IDMS DC001003 V02 T107 TASK:RHDCNP3S PROG:XXXOL100 STALLED WAITING FOR LMGR 8000 40D1C1
3.48.48 JOB08303 +IDMS DC027007 V02 T107 TASK:RHDCNP3S PROG:XXXOL100 ABENDED WITH CODE D002
3.48.48 JOB08303 +IDMS DC201006 V02 T1 CV-Status BE-TaskID Pri FE - ID1 FE - ID2 FE TaskCD FE UserID
3.48.48 JOB08303 +IDMS DC201006 V02 T1 ABRT D002 107 100 CICSBULK CICS C607 L100 DEF456
```

Figure 5 - Stalled Waiting - Case Study #1

How do we interpret the above? What is the problem?

1) Notice that program XXXGL240 is “stalled waiting on LMGR”. LMGR stands for “lock manager”, and is always followed by a code and a DBKEY or page number.

2) Use a hex calculator or program to convert the page-number (or DBKEY)

X'40D1C1' = decimal 4248001

If a DBKEY, then you usually divide by 256 (depends on your DBKEY “radix”) to find the page number, for example:

X'40D1C502' = decimal 1087489282 / 256 = 4248005.007813

so the page number would be 4248005 and the remainder (1087489282 - 256*4248005) = 2

Or convert this way - break the X'40D1C502' into two parts: X'40D1C5' and X'02'.

The first converts to 4248005 and the second converts to 2.

Thus the DBKEY = 4248005:2

- 3) Find out the area-name that contains this page number.
 - a) contact your DBA
 - b) do a DCMT DIS AREA command
 - c) run an IDMSLOOK or IDMSRPTS against your global DMCL

Sample of a DCMT DIS AREAS command

SEG1.JOB-DEMO-REGION	Upd	4236001	4248000	0	0	0	0
Stamp: 1997-05-26-16.23.20.987594		Pg grp: 0	Stat:	0	Pnd Lock:		
SEG1.EMP-DEMO-REGION	Upd	4248001	4864000	6	0	0	0
Stamp: 1998-03-29-10.22.59.253082		Pg grp: 0	Stat:	0	Pnd Lock:		
SEG1.ORG-DEMO-REGION	Upd	4900001	4913330	0	0	0	0
Stamp: 1999-02-21-15.22.25.667865		Pg grp: 0	Stat:	0	Pnd Lock:		

Figure 6 -Sample DCMT DIS AREAS

From this listing, page 4248001 is the first page of the EMP-DEMO-REGION.

When this is the first page of an area, it is likely that you are dealing with an area lock, i.e. a job is running with the area in PROTECTED UPDATE or EXCLUSIVE UPDATE (or even PROTECTED or EXCLUSIVE RETRIEVAL mode). This means that the READY statements in the program have specified that other programs needing this area will go into a wait. The online program will wait up to the External Wait Time (as specified in the SYSGEN), then it will ABEND with this message. While these programs are waiting, they are taking up memory in the program and storage pool, which often results in SOS (Short On Storage) conditions. If dozens of online programs are started from CICS, they are treated as External Run Units also, and soon the system reaches MAX TASKS or MAXERUS (maximum external run units). Then new jobs that try to start will get a 1473 (see page 37).

Catch 22 - It would be nice to the External Wait Time low, so these programs would ABEND quicker and free up resources and prevent you reaching the MAX TASKS or MAXERUS (external run units) conditions. However, External Wait Time also applies to batch jobs. When a batch job waits 21 minutes for a tape mount, it can cause the batch job to ABEND if the EXTERNAL WAIT TIME is set to 20 minutes. Thus, the External Wait Time generally needs to be set high for batch jobs.

DC001000 - Case Study #2 (Deadlock)

Deadlock Example:

10.47.52	JOB08303	+IDMS	DC001000	V02	T13	T:000102459	A120	P:A120	C:DEAD	WAITING	ON	R:LMGR	LK	00010000	C3A3D347
10.47.52	JOB08303	+IDMS	DC001000	V02	T13	T:000102477	B444	P:B444	C:DEAD	WAITING	ON	R:LMGR	LK	00010000	C29D1001
10.47.52	JOB08303	+IDMS	DC001002	V02	T13	T:000102477	B444	P:B444	C:DEAD	DEADLOCKED	ON	R:LMGR	LK	00010000	C29D1001

Figure 7 - Deadlock - Case Study #2

1) Two tasks were waiting on each other (A120 and B444) and B444 was terminated

2) The 00010000 after LMGR LK is interpreted as follows:

- The first four characters 0001 is the page group. Many sites do not use page groups, so you will see all zeroes here.
- The last four characters 0000 is the LMGR code as explained in DC001000:
 - yyyy=0000 means a page lock and zzzzzzzz is the page number.
 - yyyy=2000 means a dbkey lock and zzzzzzzz is a dbkey,
 - yyyy=8000 means an area lock and zzzzzzzz is the first page in the area,

So 0000 means that the following field (C3A3D347) is a page number.

3) So convert the page number to a humanly readable DB-KEY giving 12,821,459:071
(page: x'C3A3D3' converts to 12821459 and line-index x'47' converts to 71)

Then convert the other page number to a DB-KEY giving 12,754,192:001
(page: x'C29D10' converts to 12754192 and line-index x'01' converts to 01)

4) Conclusion: Task A120 was waiting on DB-KEY 12,821,459:071 which was held by B444.
B444 was waiting on DB-KEY 12,754,192:001 which was held by A120.
The IDMS lock manager detected this problem and terminated B444.

DC027007 - Case Study #3 (ABRU)

```
15.03.59 JOB08303 +IDMS DC027007 V02 T275850 TASK:RHDCNP3S PROG:GSFND510 ABENDED WITH CODE ABRU
15.03.59 JOB08303 +IDMS DC201006 V02 T1 CV-Status BE-TaskID Pri FE - ID1 FE - ID2 FE TaskCD FE UserID
15.03.59 JOB08303 +IDMS DC201006 V02 T1 ABRT CKUR          275850 100 BATCBULK GSLWHD15 GSWHD510 BATUSER
```

Figure 8 - ABRU Case Study #3

The ABRU message is the “catch all” when the IDMS-STATUS paragraph is performed and the IDMSCANC routine is called. You will need to look at the job’s SYSOUT for more details - as shown in Figure 11.

IDMS ERROR-STATUS

General Format

Normal IDMS programming results in a four character error code being returned from each call to IDMS (on any DML verb such as OBTAIN, MODIFY, STORE, BIND, READY, etc...)

The format is

xyyy

where xx is the Major Code and yy is the Minor Code.

EXAMPLE: 1205

The major code is 12 - for a STORE verb. (See table of Major Codes on this page)

The minor code is 05 - meaning DUPLICATES not allowed.

A list of minor codes and recommended solutions can be found from page 25 to page 41.

Each Major-Code represents an IDMS DML Verb

Major Code	DML Function
00	Any DML Function
01	FINISH
02	ERASE
03	FIND/OBTAIN
05	GET
06	KEEP
07	CONNECT
08	MODIFY
09	READY
11	DISCONNECT
12	STORE
14	BIND
15	ACCEPT
16	IF
17	RETURN
18	COMMIT
19	ROLLBACK
20	LRF Requests

Figure 9 - IDMS Error-Code - Major Codes

A detailed explanation of Minor Codes can be found from pages 25 through 41 of this booklet.

Identifying IDMS errors in Job Output (SYSOUT)

1. If ABEND U2222, look for error-status
(do the following FIND command: "F ABORTING")
2. Look for other IDMS DB messages in the output

```
J E S 2   J O B   L O G   --   S Y S T E M   G S L P   --   N O D

10.15.37 JOB05801 IRR010I  USERID NRW      IS ASSIGNED TO THIS JOB.
10.15.38 JOB05801 ICH70001I NRW      LAST ACCESS AT 10:13:47 ON WEDNESDAY, JULY
10.15.38 JOB05801 $HASP373 NRWDTSSTE STARTED - INIT  12 - CLASS N - SYS GSLP
10.15.38 JOB05801 IEF403I  NRWDTSSTE - STARTED - TIME=10.15.38
10.16.45 JOB05801 +IDMS DB347011 dbname XXXXXXXX invalid - binding subschema is
+***** ABORTING - TESTERR1, 1477,
10.16.46 JOB05801 +IDMS RUN-UNIT CANCELLED DUE TO PROGRAM REQUEST
10.16.46 JOB05801 IEA995I  SYMPTOM DUMP OUTPUT
USER COMPLETION CODE=2222
TIME=10.16.45  SEQ=04261  CPU=0000  ASID=0031
PSW AT TIME OF ERROR 078D1000 851000DE ILC 2 INTC 0D
ACTIVE LOAD MODULE          ADDRESS=05100080 OFFSET=0000
NAME=IDMSCANC
DATA AT PSW 051000D8 - 00181610 0A0D1814 0A0D0700
GPR 0-3 80000000 800008AE 0000CB08 00006D60
GPR 4-7 000008AE 0004B298 051000B4 851000A4
GPR 8-11 80012BE0 0000A6A0 0004ACC0 851000A4
GPR 12-15 80012662 0004AFE8 80012BDE 00000000
END OF SYMPTOM DUMP
10.16.46 JOB05801 IEC130I  SYSABOUT DD STATEMENT MISSING
10.16.46 JOB05801 +IGZ043I  A 'SYSABOUT' error occurred. The ABEND information
10.16.46 JOB05801 +      incomplete.
10.16.46 JOB05801 +IGZ057I  An ABEND was intercepted by the COBOL run-time ABEN
10.16.46 JOB05801 +      It is described by a corresponding IEA995I message.
10.16.46 JOB05801 IEF450I  NRWDTSSTE STEP01 - ABEND=S000 U2222 REASON=00000000
TIME=10.16.46
10.16.46 JOB05801 *END STEP STEP01 OF NRWDTSSTE TIME 10:16      **** ABEND U2222
10.16.46 JOB05801 IEF404I  NRWDTSSTE - ENDED - TIME=10.16.46
10.16.46 JOB05801 JOB NRWDTSSTE END DATE 96.213 CPU 0.002 75085AA 8000XXXXA TIM
10.16.46 JOB05801 $HASP395 NRWDTSSTE ENDED
----- JES2 JOB STATISTICS -----
31 JUL 1996 JOB EXECUTION DATE
379 CARDS READ
```

Figure 10 - Sample Abend Listing on SDSF

TESTERR1 (the word after ABORTING") is the problem program name. 1477 is the error-status. A related IDMS DB message is "IDMS DB347011 dbname xxxxxxx invalid."

Additional IDMS Diagnostic Information in Job Output

**If ABEND U2222, after finding error-status,
look for other IDMS diagnostics at bottom of //SYSOUT.**

(NOTE: This is a different error than the one shown on the prior page.)

```
RUN-UNIT BOUND, READY TO BIND EACH RECORD
RECORDS BOUND
READY TO READY AREAS:
AREA DDLCAT READIED
AREA DDLCATX READIED
ALL READIES HAVE COMPLETED
----- RETRIEVE AREAS -----          <--- Above, Misc. displays of program
PROGRAM NAME ----- TESTERR1          <--- All U2222 Abends print out this
ERROR STATUS ----- 0301              <--- information block.
ERROR RECORD ----- EMP-REC
ERROR SET ----- .
ERROR AREA ----- EMP-DEMO-REGION
LAST GOOD RECORD --
LAST GOOD AREA ----
DML SEQUENCE-----0000000008
```

Figure 11 - IDMS Standard Error Block

The DML Sequence refers back to a DML verb in the program. This is primarily useful to the programmer, but a compile listing of the program is necessary to tie this number back to the COBOL DML statement.

COBOL CODE: Finding DML-SEQUENCE and Error Handling Logic

Example of DML-Sequence and PERFORM IDMS-STATUS routine in a COBOL program:

```

                *IDMS-CONTROL SECTION.
                *PROTOCOL.  MODE IS BATCH DEBUG.
                *              IDMS-RECORDS MANUAL.

NOTE: DML-SEQUENCE numbers are ONLY generated when you have the DEBUG clause
      in the PROTOCLL / MODE IS statement!

000488          2021-NEXT-EMPOSITION.
000489          *   OBTAIN NEXT EMPOSITION WITHIN EMP-EMPOSITION      <---This DML VERB
                                                expanded to an call IDMS statement.

000490          MOVE 8 TO DML-SEQUENCE
000491          CALL 'IDMS' USING SUBSC HEMA-CTRL
000492          IDBMSCOM (10)
000493          SR420
000494          EMP-EMPOSITION
000495          IDBMSCOM (43);
000496          IF DB-END-OF-SET      <--- Program test for expected conditions
000497          1   GO TO 2021-EXIT
000498          END-IF
000499          PERFORM IDMS-STATUS
000500          ADD 1 TO WS-EMP-COUNTER
000501          ... program code continues ...

IDMS-STATUS is a module of source code usually stored in the IDMS data dictionary (IDD).
It is copied into every program, and controls the final displays, ROLLBACK, and ABORT
when the ERROR-STATUS is non-zero.
*COPY IDMS IDMS-STATUS.
*****
IDMS-STATUS
*****
IDMS-STATUS-PARAGRAPH.
    IF DB-STATUS-OK GO TO ISABEX.
    PERFORM IDMS-ABORT.
    DISPLAY '*****'
          ' ABORTING - ' PROGRAM-NAME
          ', '          ERROR-STATUS
          ', '          ERROR-RECORD
          ' **** RECOVER IDMS ****'
    UPON CONSOLE.
    DISPLAY 'PROGRAM NAME ----- ' PROGRAM-NAME.
    DISPLAY 'ERROR STATUS ----- ' ERROR-STATUS.
    DISPLAY 'ERROR RECORD ----- ' ERROR-RECORD.
    DISPLAY 'ERROR SET ----- ' ERROR-SET.
    DISPLAY 'ERROR AREA ----- ' ERROR-AREA.
    DISPLAY 'LAST GOOD RECORD -- ' RECORD-NAME.
    DISPLAY 'LAST GOOD AREA ---- ' AREA-NAME.
    DISPLAY 'DML SEQUENCE ----- ' DML-SEQUENCE.
*
    ROLLBACK.
    MOVE 13 TO DML-SEQUENCE
    CALL 'IDMS' USING SUBSCHEMA-CTRL
          IDBMSCOM (67).
    CALL 'ABORT'.
ISABEX. EXIT.
IDMS-ABORT SECTION.
IDMS-ABORT-EXIT.
EXIT.
```

Figure 12 - Finding the DML-Sequence Number

Free Error Helps Utility

Visit the <http://www.ItDoesMoreStuff.com> web site for the following free utility. It is a COBOL program that intercepts IDMS errors, and interprets them in “plain English” (with explanations similar to the ones in this document).

```
PROGRAM NAME ----- TESTERR1
ERROR STATUS ----- 1477
ERROR RECORD -----
ERROR SET -----
ERROR AREA -----
LAST GOOD RECORD --
LAST GOOD AREA ----
DML SEQUENCE-----000000008
-----
- IDMS ABEND - EXPLANATION OF YOUR ERROR-STATUS - <---- New messages start here
-----
IDMS-ERROR-STATUS=1477
MAJOR=BIND <---- Tells you the DML Verb
MINOR=RUN UNIT HAS NOT BEEN BOUND (OR BOUND TWICE) <---- and short description of minor code
[The above two lines are also written to
the console.]
<-----followed by long description
DESCRIPTION:
1) A DML STATEMENT HAS BEEN ISSUED BEFORE
THE REQUIRED INITIAL "BIND RUN-UNIT"
STATEMENT /
2) MORE THAN ONE "BIND RUN-UNIT"
STATEMENT HAS BEEN ISSUED
3) NO "BIND RUN-UNIT" STATEMENT WAS
ISSUED AFTER A "FINISH" OR "ROLLBACK"
BUT OTHER DML COMMANDS WERE ATTEMPTED.
-----
END OF PROGRAM: IDMSERRD
-----
WHEN-COMPILED=02/26/9911.23.14 <--- I also show you a trick to get these
WHERE-COMPILED=CV05/GSLDICT <--- fields to display.
```

Figure 13 - Free Utility - Enhanced Error Reporting

Handling of Most Common Error Statuses

The following three pages show the most common combinations of MAJOR/MINOR codes, and who normally has to be called to fix the problem.

Common Errors Correctable by Database Administration:

Major	Minor	Description	Action
09	66	DBA varied database offline for a reason.	DBA may be doing a reorg or maintenance on an IDMS area.
nn	11	Insufficient Space	Time must be schedule to expand the database.
03 nn	01 01	Required Area Not Readied	Likely relates to separating an index out from the data area and the new index area was not added to the program or the program did not recompile correctly.
09	71	Page range doesn't match DMCL	DMCL or Segment Problem

Figure 14 - IDMS Common Errors - Corrected by DBAs

Common Errors Correctable by Operations:

Major	Minor	Description	Action
nn	69	Run Unit Could not Start or Contention Problem See full explanation of this error on page 35.	Try to rerun job again in 5 or 15 minutes. Check CV log for abend in module IDMSDBMS or other critical system module - and if found, call DBA.
14	69	CV not active (down or in process of shutting down, or too many run units active) See full explanation of this error on page 35.	Bring up CV and run job again or wait 5 minutes and submit job again.
09	66	Area not Available in desired Usage Mode. Area was varied offline either for a batch local job to run, or by the DBA to prevent updates. Must look at sysout to find problem area-name.	Check to see if any batch local job are running. If so, wait till local job finishes and re-submit Otherwise call DBA.
nn	29	Deadlock - caused by two or more programs locking the same records.	Allow one of the programs to finish, then rerun the program that failed.

Figure 15 - IDMS Common Errors - Correctable by Operations

Common Errors Correctable by Programming:

Major	Minor	Description	Action
nn	05	Duplicate Record	Program should check for duplicates before doing store or modify.
nn	09	Program should have readied area for update but did not.	Add "Ready Area" statement to program and recompile.
nn	11	Insufficient Space in Database Program may have been in a loop or stored too many records.	Do a 3.4 to look at IDMS.DU* (IDMS DUMP data sets) to see area utilization. Check to see if program went in a loop storing records.
nn	18	Record not bound	Add Bind Record statement to program and recompile.
nn	20	Current record of Run Unit not same as the record named in DML	
03	26	Record Not Found	Bad key or desired record not on database. The program should ALWAYS check for and handle this type of error.

Figure 16 - IDMS Common Errors - Correctable by Programming

Detailed Error Messages - by Minor Code

major	minor	Summary/Suggested Action	Possible Causes
nn	00	0000 indicates successful completion but nn00 indicates problems connecting to the CV	1) The "BIND RUN-UNIT" statement or "BIND RECORD" statement has not been issued or cannot be recognized. This usually means that the IDMS Communications Block (SUBSCHEMA-CTRL) is not aligned on a fullword boundary.
nn	01	An AREA has NOT been READIED or 1601 = Empty Set	1) The area in which the object record participates has not been readied. Probably causes are: (a) No "READY" statement has been issued since the most recently issued "BIND RUN-UNIT" statement. (b) The "READY" statement is missing or has been bypassed by program logic. 2) For a '1601' - IF STATEMENT - this means that the named SET is empty, or the record that is current of run-unit is not a member of the named set. The status triggers or bypasses execution of the imperative statement.
nn	02	DBKEY is inconsistent	1) DBKEY may have a wrong value, nulls, zero, or overlaid working storage 2) Record name is incorrect 3) Wrong Subschema 4) DBKEY is outside of page range for Area
nn	04	Bad value in OCCURS DEPENDING ON	1) Look for an "occurs depending on" in the IDMS record layout. 2) Might be less than zero or greater than maximum 3) Wrong Subschema

major	minor	Summary/Suggested Action	Possible Causes
nn	05	Duplicate Key	1) Check the CALC key 2) Check all sort keys and index sets 3) Add program code to check for duplicates before attempting to do a STORE, CONNECT, or MODIFY that might violate a DN (Duplicates not allowed option).
nn	06	No Currency Established	You attempted to do an “OBTAIN NEXT”, “OBTAIN OWNER”, “OBTAIN DUPLICATE” “KEEP”, “CONNECT”, or “DISCONNECT” without first getting current on some record in the database.
nn	07	DB-END-OF-SET	This is the “normal” error-status to be tested for end-of-set. 1) You attempted to obtain one past the last record in the set or area. 2) The set (or area) is totally empty. NOTE: This error should not usually cause an ABEND. You should check for this error and handle it appropriately within your program code following each “OBTAIN NEXT” or “OBTAIN PRIOR” command.

major	minor	Summary/Suggested Action	Possible Causes
nn	08	<p>The named record or set is not in the specified subschema.</p> <p>Program may have invoked the wrong subschema, the record or set name may have been overlaid.</p>	<p>1) The object record (or setname) is not in the specified subschema (wrong record-name or record removed from subschema since the program was compiled).</p> <p>2) The named record is not defined as a member of the named set.</p> <p>3) Double check that the record or set name in storage was not overlaid (perhaps by a subscribing problem).</p>
nn	09	<p>Tried to UPDATE when AREA READIED in RETRIEVAL</p>	<p>1) You tried to do a command that would update the database (modify, erase, store, connect, or disconnect) but the database area was not readied for update. (Add the "UPDATE" parm to your READY statement.)</p> <p>2) Maybe a recent database RE-ORG added new areas to the database (for instance, maybe the index records were moved to their own area-name, and that area now also needs to be readied in update mode.)</p>
nn	10	<p>SUBSCHEMA ACCESS RESTRICTION</p>	<p>The subschema in use has an access restriction against this type of DML verb.</p>

major	minor	Summary/Suggested Action	Possible Causes
08/16	13	Run-unit currency not established	1) Run-Unit Currency has been nullified by a previous ERASE statement 2) Run-Unit Currency has not been established.
11	15	Cannot Disconnect from a Set with Options of MA or MM	1) Cannot DISCONNECT a record from a set where the first connect option is "MANDATORY" 2) The schema "SET OPTIONS" must be changed if you need this capability.
07	16	Records is already a member of set	The record is already a member of the set, and therefore cannot be connected to the set. NOTE: Use the "IF SET MEMBER" statement or DISCONNECT the record before attempting the connect.
nn	18	The record has not been bound.	A BIND statement must be issued before the current verb is issued.

major	minor	Summary/Suggested Action	Possible Causes
nn	21	Not all areas being updated have been readied in UPDATE usage mode or not all records, sets, and areas affected have been included in subschema.	The most probable cause is that the wrong subschema has been invoked or that the subschema has been compiled incorrectly. If the object record is a member of a multi-member set, there are two additional causes for this code: (1) an area in which one of the other members resides has not been readied in update mode (2) one or more of the other members of a multi-member set have not been included in the subschema.
11 Disc	22	The named record is not currently a member of the named set.	You are attempting to DISCONNECT a record from a set from which that record is not a member. NOTE: Use the "IF SET MEMBER" command first to see if the record is a member of the set.
nn	23	The named area is not in the subschema.	The database may have been redesigned (re-orged) and this program was not updated to reflect the new design.
12	25	No currency established for the object set.	Need to obtain currency on the owner record before storing a member record (in a mandatory automatic set).

major	minor	Summary/Suggested Action	Possible Causes
03	26	Record not found or no more duplicates exist for this record.	1) Usually the program traps this message and does its own error handling. The "0326" code is associated with the 88-level "DB-REC-NOT-FOUND". 2) Wrong, misspelled, or corrupted CALC KEY or index sort key 3) The record has been erased from the database 4) Might also be caused by a change in the base VDB vs. a VDB shadow file.
02	26	Record not found	Possible Broken Chain. If you are running on a non-production system, this error often means a VDB shadow needs to be refreshed
03/17	26	Record not found	The indexed entry cannot be found. The prior of set and next of set is set to null. A subsequent format 3 FIND/OBTAIN statement specifying the next or prior of set will be executed properly. I.E. currency has been set even though the exact record key was not found.
nn	28	The run unit attempted to ready an area that has already been readied.	1) You must do a "FINISH" or "ROLLBACK" before re-readying an area within the same program. 2) Maybe calling program readied the area and the program is being called as an extended-run-unit.

major	minor	Summary/Suggested Action	Possible Causes
02	30	An attempt has been made to ERASE the owner record of a NON-EMPTY SET	The unqualified form of the ERASE verb has been issued, but the record occurrence specified is the owner of a non-empty set. The program should either do an "ERASE ALL" or should manually walk the set and erase each of the member records - or at least make sure that there are no member records before attempting the unqualified erase.
03	31	The format of the retrieval statement conflicts with the record's location mode (CALC, VIA, or DIRECT)	1) FIND/OBTAIN CALC has been used to retrieve a record that does not have a location-mode of "CALC" 2) FIND/OBTAIN within a set using SORTKEY was used to retrieve a record within a set that is NOT sorted.
03 Obt.	32	An attempt to retrieve a DUPLICATE CALC record was unsuccessful;	FIND/OBTAIN DUPLICATE can only be issued when the current record (of record-type) has a CALC-KEY that matches the CALC-KEY contained in the program's working storage.
02 Erase 08 Mod 12 Store	33	Set(s) not in subschema	One or more sets in which the record participates have not been included in the subschema, which caused the subschema compiler to establish access restrictions when the subschema was compiled.

major	minor	Summary/Suggested Action	Possible Causes
03 Obt	52	IDMS attempted to read an incomplete variable-length record.	In a system with "RETRIEVAL NOLOCK" IDMS attempted to retrieve a variable-length record for which a "STORE" operation was still in progress.
nn	54	Invalid record description (SR51) in subschema tables.	The prefix length of an SR51 is less than zero or the data length is less than or equal to zero.
03 Obt	56	Insufficient Memory for Compression/Decomp. Subroutines	See Database Administrator about increasing size of the storage pool.
nn	60	A record occurrence has been encountered whose type is inconsistent with the set named in the error set field of the IDMS Control Block.	1) An affected record occurrence has been found in the database, but its record-type has not been found in the subschema as a member or owner of the set. Probable causes are: (a) A broken chain exists in the database due to improper recovery. (b) A new schema and associated subschemas describe an existing database improperly. (c) VDB Shadow (Virtual Database) too far out-of-sync with live database.

major	minor	Summary/Suggested Action	Possible Causes
nn	63	The DBMS cannot interpret the DML function to be performed	<p>1) A direct use of the (hard-coding) the "CALL IDMS" statement (rather than using DML) was not coded correctly (had invalid parameters).</p> <p>2) A PL/I program has not declared the IDMS interface as an assembler program.</p> <p>3) An attempt has been made to perform a CA-IDMSDC function from a DC-BATCH program but the function is not supported in DC/BATCH (such as accessing DC/UCF scratch/queues or reports (WRITE PRINTER)).</p>
nn	64	The CALC control element has not been described properly.	The named record has not been included in the subschema with a CALC location mode, but the CALC control element has not been described properly in the subschema.
nn	65	<p>The database page read was not the page requested.</p> <p>Often a 1265 on the reload utility.</p>	<p>A page read into the system buffers from disk contains first and last 4-byte positions that do not equal the page number requested.</p> <p>1) IDMS INIT/Format Utility not run for area</p> <p>2) JCL or segment points ddname to wrong data set name</p> <p>3) page size has been specified incorrectly (or file allocated with wrong blksize?)</p> <p>4) expand-page was run on area, but DMCL/segment not updated to reflect this</p> <p>5) data on page has been destroyed by a non-IDMS program or utility</p> <p>6) check for I/O error</p>

major	minor	Summary/Suggested Action	Possible Causes
nn	67	The subschema invoked on a BIND RUN UNIT is an invalid subschema load module or the internal name in the subschema load module does not match the specified subschema name.	<ol style="list-style-type: none"> 1) The tables loaded are not subschema object tables. 2) Load Libraries (//STEPLIB or //JOB LIB) have been incorrectly concatenated 3) Subschemata was linked with the wrong name 4) An attempt has been made to use an alias through the MVS linkage editor. This is not allowed for subschemata. 5) The subschema was added to the IDMS dictionary load area using the wrong name.
nn	68	The CICS Interface (IDMSINTC) was not started.	The INTC task is normally automatically started by CICS when it comes up. Could be corruption in CICS CWA or a bad system definition by the DBA.

nn	69	Usually 0069 or 1469 The run unit has lost communications with the IDMS CV.	<p>1) The CV is not active, or the CV went down while your job was running.</p> <p>2) The CV is not accepting new run units (at max-tasks)</p> <p>3) The program is not registered</p> <p>4) Proper BIND statement not issued by the program.</p> <p>5) Maximum external wait time has been exceeded (too long between calls to IDMS, such as CPU loop, waiting on tape or other resource). "EXTERNAL WAIT" time is defined in the SYSGEN.</p> <p>5) The subschema-control block could have been overlaid (double check for array/subscribing problems and or subroutine linkage problems)</p> <p>6) a nn29 or nn79 error status has been encountered (ending the run unit), but the program failed to check for (or properly handle) these status values</p> <p>7) <u>Look at the CV log for further associated messages!</u> Often a D002, D003, or other 4 digit serious IDMS ABEND code will show-up (such as an 1143 for a corrupt index). See overview of IDMS Abend Codes on page 8 of this document.</p> <p>8) VDB VSAM shadow file could be full.</p>
----	----	--	---

major	minor	Summary/Suggested Action	Possible Causes
03 Obt 09 Ready 14 Bind Run Unit	70	File will not open properly. DMCL page size does not match physical block-size of data set on disk	1) DMCL page size does not match physical block-size of data set on disk 2) Wrong DDNAME or data set name in the JCL of a local mode job or utility (i.e. a valid ddname may point to an existing data set name, but one that is not valid for that ddname). 3) A journal file will not open properly. In a local mode job, the journal DD statement might be missing. 4) For NATIVE VSAM files - check that the cluster was properly defined using IDCAMS and that the "control interval contains" clause was correctly specified in the DMCL. May also be insufficient storage (REGION SIZE) for VSAM control blocks to be loaded.
09	71	Bad DBKEY/page The page range/page group for the area being readied, or the page range/page group requested cannot be found in the DMCL.	1) The page range or page group specified will not map properly with the DMCL provided. Check the //SYSIDMS to see if using an incorrect DMCL name. 2) The subschema and DMCL are mis-matched (or the DB-KEY is bad). 3) The DMCL has been regenerated, but the Central Version has not been cycled.

major	minor	Summary/Suggested Action	Possible Causes
14	73	A new run unit would exceed the MAXERUS (maximum external run-units) value, as specified in the SYSGEN.	<p>1) This means the system is very busy now (or having sever problems - see example below), and you should be able to just re-run the job again in a few minutes.</p> <p>2) The DBA's may need to change MAXERUS (maximum external run units) in the SYSGEN.</p> <p>Other than an abnormally busy system, here is an example that can cause the system to be "busy". You have a batch job running in protected update. Online transactions are trying to access the same area, so they go into a WAIT state (up to the number of seconds specified in the external wait time). So instead of finishing in 1 or 2 seconds, the stay in a WAIT state for 1 to 20 minutes. New transactions come in, and they try to access the area, and go into a wait state. So in a matter of minutes, you have 70 online tasks all active - but in a WAIT state.</p> <p>If your MAXERUS is set to 70 - you have reached it - and all new income transactions will ABEND with a 1473.</p>

nn	74	The dynamic load of a module has failed.	<ul style="list-style-type: none"> 1) Module not in library 2) A PDE (Program Definition Element) that the CV can assign to the module does not exist 3) An I/O error has occurred during the load of the module 4) The CV may need to be recycled if a loadlib was compressed. 5) A database procedure could not be found or loaded. 6) It's possible that someone compressed the load library without cycling the CV.
nn	75	A READ error has occurred.	<ul style="list-style-type: none"> 1) File Assignment has been specified incorrectly in the JCL (check spelling and DDNAME to DSN matching). 2) Physical I/O Hardware Error? 3) Disk Journal Files or Database Files have not been initialized 4) Insufficient buffer size in the DMCL 5) Unexpected return code from VSAM

major	minor	Summary/Suggested Action	Possible Causes
nn	77	The run unit has not been bound or has been bound twice.	<p>1) A DML statement has been issued before the required initial "BIND RUN-UNIT" statement.</p> <p>2) More than one "BIND RUN-UNIT" statement has been issued.</p> <p>3) No "BIND RUN-UNIT" statement was issued after a "FINISH" or "ROLLBACK" but other DML commands were attempted.</p> <p>4) The program got a different IDMS ERROR-STATUS on the prior verb, causing the run unit to terminate. The program code failed to check for this error, and allowed the program to continue, thus leaving the real cause of the error unknown.</p>
nn	78	An area wait deadlock has occurred.	To wait for an area would cause a deadlock. It is recommended that all areas be readied either before issuing the first function call (DML VERB) or that all program ready areas are in the same order.
nn	80	The target node is not active or has been disconnected.	<p>1) The node specified through the "BIND RUN-UNIT" statement is not active or has been disconnected. Note that the node may have been specified explicitly in the "BIND RUN-UNIT" statement or it may have been derived from the //SYSIDMS, IDMSOPTI, or SYSCTL or Database Name Table.</p> <p>2) The node name specified was misspelled.</p>

major	minor	Summary/Suggested Action	Possible Causes
nn	82	The subschema is not allowed under the specified database (DBNAME).	The subschema name specified in the "BIND RUN-UNIT" statement cannot be located in the database name table, and the specified database has been defined with the "ALWAYS" option.
nn	83	AN error has been made in the use of native VSAM data sets.	A native VSAM error has occurred during an attempt to access a native VSAM data set. This major code usually results from a VSAM I/O Error and is issued in conjunction with sytem runtime error message 205006.
nn	84	READY or first DML command.	The threshold for number of tries to lock an area has been exceeded. Repeated tries were made to "READY AREA" until the threshold was exceeded.
nn	86		An attempt to lock a db-key would have requested a wait but "nowait" was requested.
14	91	BIND RUN-UNIT	There must be a DBNAME established before issuing a "BIND RUN-UNIT" to the specified subschema name. NOTE: You can always use a segment-name instead of a dbname.

major	minor	Summary/Suggested Action	Possible Causes
14	93	BIND RUN-UNIT	The subschema specified contains a symbolic value for the area name specified. The symbolic value does not match a corresponding symbolic value in the DMCL.
14	94	BIND RUN-UNIT	The DBNAME specified does not match any DBNAME defined in the Database Name Table or any segment defined in the DMCL. The status also results if a DBNAME defined in the DBNAME TABLE is invalid.
14	95	BIND RUN-UNIT	The specified subschema must be defined in the default section of the Database Name Table when a bind is issued with the specified DBNAME.

IDMS Error Subroutine

Summary: IDMSERR1 and IDMSERRD are COBOL subroutines that give full descriptions of the four byte IDMS Error-Status Codes. The descriptive information can be displayed on the CONSOLE, the //SYSOUT, or just returned to the calling program. This applies to database error codes (not DC) and does not touch or impact any of the DB##### or DC##### error codes.

Availability: This subroutine can be downloaded and used as FREEWARE from www.ItDoesMoreStuff.com

Purpose: This program serves several purposes

- 1) Saves the programmer from having to look up the error status in the Error Code and Messages manual, thus improving development time
- 2) Give the operators and scheduling group more powerful and targeted error descriptions allowing for faster problem resolution and turn-around on the production schedule. Probable causes and suggested actions aid in solving the problem.
- 3) Error messages displayed by program are similar to those in manual, but contain additional information available from years of experience and messages can be custom tailored to our sites special needs (for example how VDB shadow might cause the error) and the messages can be tuned and improved in the future.

Sample //SYSOUT:

```
-----  
- IDMS ABEND - EXPLANATION OF YOUR ERROR-STATUS -  
-----  
IDMS-ERROR-STATUS=1209  
MAJOR=STORE  
MINOR=AREA NOT READIED IN UPDATE MODE  
IDEAS:  
1) YOU TRIED TO DO A COMMAND THAT  
   WOULD UPDATE THE DATABASE  
   (MODIFY/STORE/CONNECT/DISCONNECT/  
   ERASE) BUT THE DATABASE AREA  
   WAS NOT READIED FOR UPDATE.  
2) MAYBE A RECENT DATABASE RE-ORG  
   ADDED NEW AREAS TO THE DATABASE  
   (FOR INSTANCE, MAYBE THE INDEX  
   RECORDS WERE MOVED TO THEIR OWN  
   AREA-NAME, AND THAT AREA NOW NEEDS  
   TO ALSO BE READIED IN UPDATE).  
-----  
END OF PROGRAM: IDMSERRD
```

Figure 17 - Example IDMS-ERROR Interpretation //SYSOUT

The first error description above is similar to the one you would find in the IDMS ERROR CODES AND MESSAGES manuals. The second error description is the type of error that has been added after many years of experience.

14.07.08	JOB01075	ESY101I	Jobname	Stepname	Procstep	RC	EXCP	Conn
14.07.20	JOB01075	+IDMS-ERROR-STATUS=1209						
14.07.20	JOB01075	+MAJOR=STORE						
14.07.20	JOB01075	+MINOR=AREA NOT READIED IN UPDATE MODE						
14.07.21	JOB01075	ESY102I	NRWIDMSE	STEP02		00	83	166

Figure 18 - Optional LOG (Console) Messages

Subroutine: IDMSERR1

- 1) Assumes the most displays possible
- 2) Requires no additional working storage variables to be added to the program (only the IDMS ERROR-STATUS field is passed to the subroutine) .
- 3) This can be added to the standard "IDMS-STATUS" routine which is copied into all Batch Programs (CICS programs should not call this routine because it does COBOL Displays).

162200	CALL 'IDMSERR1' USING ERROR-STATUS.
--------	-------------------------------------

Figure 19 - How to call from COBOL

IDMS-STATUS is a module stored on the IDMS data dictionary. Use the command `DIS ALL MODULE WHERE NAME = IDMS-STATUS` to identify all the IDMS-STATUS modules. There will be one for each MODE, such as BATCH-AUTOSTATUS, IDMS-DC, DC-BATCH, etc... Each IDMS-STATUS will have a different version number. The IDMS DML preprocessor picks up the correct version based on the MODE statement coded in the COBOL program.

The IDMSERRD routines, as written, use the COBOL DISPLAY commands, and are thus made to work for batch only. With minimal effort, they could be adapted for DC or CICS online COBOL programs. So, DO NOT add a call to IDMSERRD to CICS or DC-COBOL online programs.

It is recommended that the new more descriptive error messages be added as shown below:

```

002514 *COPY IDMS IDMS-STATUS.
002515 *****
002516 IDMS-STATUS.
002517 *****
002518 *IDMS-STATUS-PARAGRAPH.
002519     IF NOT DB-STATUS-OK
002520         PERFORM IDMS-ABORT
002521         DISPLAY '*****'
002522                 ' ABORTING - ' PROGRAM-NAME
002523                 ' , ' ERROR-STATUS
002524                 ' , ' ERROR-RECORD
002525                 ' **** RECOVER IDMS ****'
002526                 UPON CONSOLE
002527                 DISPLAY 'PROGRAM NAME ----- ' PROGRAM-NAME
002528                 DISPLAY 'ERROR STATUS ----- ' ERROR-STATUS
002529                 DISPLAY 'ERROR RECORD ----- ' ERROR-RECORD
002530                 DISPLAY 'ERROR SET ----- ' ERROR-SET
002531                 DISPLAY 'ERROR AREA ----- ' ERROR-AREA
002532                 DISPLAY 'LAST GOOD RECORD -- ' RECORD-NAME
002533                 DISPLAY 'LAST GOOD AREA ---- ' AREA-NAME
002534                 DISPLAY 'DML SEQUENCE ---- ' DML-SEQUENCE
* IN-HOUSE CUSTOMIZATION
CALL 'IDMSERR1' USING ERROR-STATUS
MOVE WHEN-COMPILED TO DBA-WHEN-COMPILED
MOVE 'CV01/APPLDICT' TO DBA-WHERE-COMPILED
DISPLAY ' WHEN-COMPILED=' DBA-WHEN-COMPILED
DISPLAY ' WHERE-COMPILED=' DBA-WHERE-COMPILED

```

Figure 20 - Copy IDMS-STATUS Common Code

The WHEN-COMPILED and WHERE-COMPILED variables:

It is often useful to know when a program was compiled and where it was compiled. The WHEN-COMPILED is a COBOL system variable. The WHERE-COMPILE can be hard-coded for each IDMS-STATUS routine in each CV and each dictionary. This helps to know if the program was migrated from a staging dictionary, or perhaps was compiled “in-place” on the production dictionary.

The trick here is to make the variables part of the load module without requiring changes to hundreds of programs. By having this information in the load module, it is not too difficult to browse the load module and to identify these variables.

This can be accomplished by adding the two variables (DBA-WHEN-COMPILED and DBA-WHERE-COMPILED) onto the end of the SUBSCHEMA-CTRL record in each dictionary. This record is generally copied into every IDMS program and is used to pass IDMS information to the IDMS subroutine. Adding extra fields on the end of the record will not affect how the IDMS subroutine works.

Types of Error Explanations:

NOTE: There are three types of error explanations (see Figure 17 and Figure 18 above):

Example:

14.07.20	JOB01075	+IDMS-ERROR-STATUS=1209
14.07.20	JOB01075	+MAJOR=STORE
14.07.20	JOB01075	+MINOR=AREA NOT READIED IN UPDATE MODE

Figure 21 - Repeat of Figure 18

1) The Error Status Code Itself: - for example 1209

- This code is already displayed by the common “COPY IDMS-STATUS” code, but can be displayed again before the Short and Long messages. This automatically happens when calling IDMSERR1 but when calling IDMSERRD, you must test IDMS-CODE-YN-DISPLAY to “Y”.

2) The “Short” Messages (two or three of them)

- (1) MAJOR Code - indicates the Verb Name - 20 characters
- (2) MINOR Code - a short explanation of the MINOR code
- (3) (90% of them fit in one 50-character field, but occasionally a second 50-character field is required to fully explain)

- These three values are returned in the LINKAGE variables when calling IDMSERRD, but not when calling IDMSERR1

- They are also displayed to the //SYSOUT when IDMS-ERROR-YN-DISPLAY is set to “Y”.

- They are also displayed to the console when IDMS-CONSOLE-YN-DISPLAY is set to “Y”

3) The “Long” Messages

These are usually labeled “ideas”, “description”, or “probable causes”. They are a printout of all the information possible that might help someone solve the problem. These are only printed by the COBOL DISPLAY statement. They appear in the //SYSOUT and are not returned in the linkage variables. There maybe from one to twenty-five lines of these types of messages.

See Figure 17 for a sample of these “Long Messages.”

Additional Subroutine: IDMSERRD

- 1) Actually displays the error messages, or optional just returns them in linkage variables
- 2) Requires the calling program to pass the correct linkage variables
- 3) This is only for internal documentation. The only programmers that need this would be if someone was doing some kind of utility program that need to trap and print special messages.

```
156100      01  IDMS-ERROR-GROUP-2 .
156200      02  IDMS-MAJOR-MEANING      PIC X(20) .
156300      02  IDMS-MINOR-MEANING-1    PIC X(50) .
156400      02  IDMS-MINOR-MEANING-2    PIC X(50) .
156500      02  IDMS-ERROR-YN-DISPLAY   PIC X .
156600      02  IDMS-CONSOLE-YN-DISPLAY PIC X .
156700      02  IDMS-CODE-YN-DISPLAY    PIC X .
156800
157000      01  IDMS-ERROR-GROUP .
157100      02  IDMS-ERROR-CODE .
157200          05  IDMS-ERROR-CODE-MAJOR      PIC XX .
157300          05  IDMS-ERROR-CODE-MINOR      PIC XX .
...
157800          MOVE 'Y' TO IDMS-ERROR-YN-DISPLAY
157900          MOVE 'Y' TO IDMS-CONSOLE-YN-DISPLAY
158000          MOVE 'Y' TO IDMS-CODE-YN-DISPLAY
158050          MOVE ERROR-STATUS TO IDMS-ERROR-CODE .
158100          CALL 'IDMSERRD' USING IDMS-ERROR-GROUP ,
158200                                IDMS-ERROR-GROUP-2 .
```

Figure 22 - Sample Call to 'IDMSERRD'

The three YN codes should be set to the value of “Y” or “N” (blank will result in “N”).

IDMS-ERROR-YN-DISPLAY - tells the subroutine to display the error-descriptions (set to “N” if you just want to get the error-descriptions back in your program and you want to display them or handle them yourself). This parms applies to the COBOL //SYSOUT.

IDMS-CONSOLE-YN-DISPLAY -Displays the error code, and the meaning of the MINOR code and the MAJOR code on the console.

IDMS-CODE-YN-DISPLAY - tells the subroutine to display the 4 character IDMS status code.

Reporting Problems to CA and Using TCC

CA's technical support number is 800-645-3042. They typically take log your problem, assign an issue number and have someone call you back. Be sure and record your issue number!

You must know your site id when reporting a problem. A PIN# (personal identification number) can optionally be assigned to each person within a site-id. This allows CA to keep your name, phone, and fax number in their database.

CA's web site is: <http://www.cai.com>. Star-TCC (Total Client Care) is the name of their support system - which can be found at <http://support.cai.com/catotalclientcare.html>. Use of this system requires that you pre-register. Once you have a userid/password, you can jump straight to Star-TCC at <https://webtrack.cai.com/cgi-bin/logon?Action=Display>.

Issues can be logged, viewed, and updated from the above web site. There is also a "Knowledge Base" where you can search for PTFs and APARs (solutions to known problems) and for Technical Notes.

ItDoesMoreStuff Web Site

<http://www.ItDoesMoreStuff.com> (capital letters NOT required)

Neal Walters email: nwalters@itdoesmorestuff.com

or call 888-240-4515 or 972-771-4091 for more information

Free Utilities

Signup for the FREE Utility of the Month Club. Approximately every month - a new utility will be added to the web site, and documentation will be available to email subscribers.

Computer Based Training Tutorials for IDMS

Current modules include:

- 1) IDMS Fundamentals and Batch Programming (workbooks available)
- 2) Introduction to ADS (includes IDD and Online Mapping)

Tutorials under consideration:

- 1) Introduction to CULPRIT
- 2) Intermediate ADS

Also available is an online subscription to a web site with over 250 training courses. Mainframe Courses include SAS, COBOL, MicroFocus COBOL, CICS, JCL, ISP, MVS/OS390, TSO/E, REXX, Easytrieve, VSAM, DB2, Oracle, and TCP/IP. Other Professional Courses include PowerBuilder, C++, VB5.0, Windows NT (Server, Workstation, Enterprise), Unix, SQL Server 6.5, OOP with C++, JAVA, LANs, HTML, FOCUS.

IDMS Discussion Forums

Talk with your IDMS peers about how IDMS really works, tips and techniques of tools such as OLQ, CULPRIT, COBOL, IDD, Mapping, ADS, and DBA stuff. All you need is a web browser to access all discussion groups. Email notification is optional.

IDMS Job Search

The only web site dedicated to only IDMS jobs, such as IDMS COBOL programming, ADS programming, System Analysts, Database Analyst, Database Administrator, and/or System Programmers.

Computer Drawing

Sign-up for a chance to win a FREE Pentium computer. Drawing held monthly.

Index

- 3### Abends, 9**
- ABEND. *See* MVS Abends. *See* IDMS Abends
 - Definition, 7
 - Locating IDMS Error Status, 18
- ABRU
 - IDMS Abend, 9
- Applid, 5
- Archive
 - Journal. *See*
 - Log, 5
- Areas, 5
- Basics of IDMS, 5
- CBT (tutorials)
 - Information, 49
- Central Version, 5**
- CICS
 - Definition, 7
- CKUR
 - IDMS Abend, 9
- COBOL
 - Interface to Error Subroutine, 43
- Codes
 - Table of All IDMS Major Codes, 17
- Commits
 - Locking, 6
- Common IDMS Errors
 - Correctable by DBA, 22
 - Correctable by Operations, 23
 - Correctable by Programming, 24
- Common Terms, 5
- Computer Based Training
 - Information, 49
- contact information, 49
- CV, 5**
- CV Batch, 6
- CV Down
 - Local Mode Update, 6
- CV Shutdown, 6
- CV Startup, 6
- D### Abends, 8
- Data Manipulation Language
 - Definition, 7
- DBKEY
 - Wait Abend DC001003, 10
- DC/LOG, 5
- DC001003
 - IDMS Abend, 10
- DCMT
 - Definition, 6
 - UCFBATCH, 6
- Deadlock, 7
- Debug Variables
 - WHEN and WHERE Compiled, 45
- Dictionary, 5
- Discussion Forums
 - Website Information, 49
- DML
 - Definition, 7
- DML Function
 - Table of All IDMS Major Codes, 17
- DML Precompiler, 7
- Error
 - Subroutine, 42
- Error-Status
 - Description from Error Subroutine, 46
- Exclusive Locks, 6
- Finding an IDMS Error, 18
- Free Utilities
 - Website Information, 49
- Fundamentals of IDMS, 5
- Glossary, 5
- http
 - [//www.itdoesmorestuff.com](http://www.itdoesmorestuff.com)
 - Website Information, 49
- IDMS Abend
 - 0069, 35
 - 0226, 30
 - 0230, 31
 - 0233, 31
 - 0307, 26
 - 0308, 27
 - 0326, 30
 - 0331, 31
 - 0332, 31
 - 0352, 32
 - 0356, 32
 - 0364, 33
 - 0370, 36
 - 0716, 28
 - 0813, 28
 - 0816, 28
 - 0833, 31
 - 0970, 36
 - 0971, 36
 - 1115, 28
 - 1233, 31
 - 1469, 35
 - 1470, 36
 - 1473, 37
 - 1491, 40
 - 1493, 41
 - 1494, 41
 - 1495, 41
 - 1726, 30
 - nn00, 25
 - nn01, 25
 - nn02, 25
 - nn04, 25

- nn05, 26
- nn06, 26
- nn07, 26
- nn09, 27
- nn10, 27
- nn18, 28
- nn21, 29
- nn22, 29
- nn23, 29
- nn28, 30, 31
- nn33, 31
- nn52, 32
- nn54, 32
- nn56, 32
- nn60, 32
- nn63, 33
- nn67, 34
- nn68, 34
- nn69, 35
- nn70, 36
- nn71, 36
- nn73, 37
- nn74, 38
- nn75, 38
- nn77, 39
- nn78, 39
- nn80, 39
- nn82, 40
- nn83, 40
- nn84, 40
- nn86, 40
- nn91, 40
- nn93, 41
- nn94, 41
- nn95, 41
- IDMS Abends
 - 3###, 9
 - ABRU, 9
 - CKUR, 9
 - D### Format, 8
 - DC001003, 10
- IDMS Areas, 5
- IDMS Basics, 5
- IDMS Discussion Forums
 - Website Information, 49
- IDMS Error Subroutine, 42
- IDMS Errors
 - Locating in a Job Abend, 18
- IDMS Jobs
 - Website Information, 49
- IDMS Major Codes
 - Table of All Major Codes, 17
- IDMS Tutorials
 - Information, 49
- IDMSERR1
 - Error Subroutine, 43
- IDMS-STATUS
 - Changes for IDMSERR1 Subroutine, 44
- Inquiry Locks, 6
- itdoesmorestuff.com
 - Website Information, 49
- Job Search
 - Website Information, 49
- Jobs (IDMS)
 - Website Information, 49
- Journals, 5
- Local Mode Update, 6
- Locating an IDMS Error, 18
- Lock. *See* also Deadlock
- Locks
 - Select or Inquiry, 6
 - Update (Exclusive), 6
- Logs, 5
- Major Codes
 - Table of All IDMS Major Codes, 17
- Migration, 7
- Neal Walters
 - contact information, 49
- Offline
 - Area Status, 5
 - Local Mode Update, 6
- Online
 - Area Status, 5
- Page Number
 - Wait Abend DC001003, 10
- PerfMon
 - Definition, 6
- Performance Monitor, 6
 - Definition, 6
- PMRM. *See* PerfMon
 - Definition, 6
- Precompiler DML, 7
- Preprocessor DML, 7
- Protected Update, 6
- Restartable Jobs
 - Locking, 6
- Restore Datasets
 - Local Mode Update, 6
- Retrieval
 - Area Status, 5
 - Local Mode Update, 6
- Rollback
 - Definition, 7
 - Journal, 5
- Select Locks, 6
- Shared Update (CV), 6
- Staging, 7
- Stall
 - DC001003 IDMS Abend, 10
- Subroutine
 - IDMS Error, 42
- SYSOUT

Additional IDMS Error Handling Data, 19
Table of All IDMS Major Codes, 17
Terminology, 5
Tutorials
 Information, 49
U2222
 Locating IDMS Error Status, 18
UCF/Batch, 6
Update
 Area Status, 5
 Shared (CV), 6
Update Locks, 6
Utilities
 Website, 49
Variables

 WHEN and WHERE Compiled, 45
VDB Definition, 7
Verb Codes
 Table of All IDMS Major Codes, 17
VTAM Applid, 5
Wait
 DC001003 IDMS Abend, 10
Web Site
 Information, 49
WHEN-COMPILED
 Debug Variable, 45
WHERE-COMPILED
 Debug Variable, 45
www.itdoesmorestuff.com
 Website Information, 49