

# CA IDMS™ Online

## DML Online User Guide

Release 18.5.00, 2nd Edition



This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by CA at any time. This Documentation is proprietary information of CA and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA.

If you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2014 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

## CA Technologies Product References

This document references the following CA products:

- CA IDMS™/DB
- CA ADS™
- CA IDMS™/DC
- CA IDMS DMLO
- CA IDMS

## Contact CA Technologies

### Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At <http://ca.com/support>, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

### Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to [techpubs@ca.com](mailto:techpubs@ca.com).

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at <http://ca.com/docs>.

## Documentation Changes

The following documentation updates were made for the 18.5.00, 2nd Edition release of this documentation:

- [Severity Codes](#) (see page 125)—Added the message E2051 KEYWORD MISSING.

The following documentation updates were made for the 18.5.00 release of this documentation:

- [Using FIND/OBTAIN Commands](#) (see page 62)—Updated the commands used to assign a value to PAGE-INFO to include ACCEPT KEYn FROM ... CURRENCY.
- [CA IDMS DML Save Statements](#) (see page 66)—Updated ACCEPT KEYn FROM ... CURRENCY syntax to include the PGRn keyword, which allows for use of a page info identifier.
- [CA IDMS DML Command Restrictions](#) (see page 73)—Updated the ACCEPT and RETURN section to include a description of the page-info identifier.
- [Severity Codes](#) (see page 125)—Added the message E5019 ERROR RETRIEVING PAGEINFO.
- [User Exit Module](#) (see page 122)—Added the information that USDMLXIT is dynamically loaded.

# Contents

---

## Chapter 1: General Information 11

What Is CA IDMS DMLO? .....	11
How Can CA IDMS DMLO Improve Productivity? .....	12
Is CA IDMS DMLO Easy to Use? .....	12
CA IDMS DMLO Features .....	13
Multiple/Distributed Database Support .....	13
Multiple Teleprocessing Environment Support .....	13
Support of CA IDMS DML Commands .....	13
Support of CA IDMS LRF Commands.....	13
Support of CA IDMS/DC SCRATCH/QUEUE Commands.....	13
Access to Dictionary Records.....	14
Extensive Set of CA IDMS DMLO Commands .....	14
Session PROFILES.....	14
CLIST Programming.....	15
Comprehensive Online Documentation.....	15
Data Manipulation and Display.....	15
PF Key Usage.....	16
EQUATE Facility .....	16
Selection Lists.....	16
Macros and Variables.....	16
Access to CA IDMS Statistics .....	17
Extensive DBKEY Capabilities .....	17
Menu/Assist Mode.....	17
Access Security .....	18
Installation Customization.....	19
Mixed Page Group Support .....	19
Enter Key Usage.....	20
User Exit Module.....	20

## Chapter 2: CA IDMS DMLO Session 21

Overview.....	21
Text Split Command.....	21
Text Flow Command.....	22
Text Entry Command.....	22
Destructive Line Shift Examples.....	23
Protective Line Shift Command .....	24

---

Introduction to the CA IDMS DMLO Session .....	24
CA IDMS DMLO Session.....	25
Invoking CA IDMS DMLO .....	26
PROFILE Selection/Maintenance.....	30
Subschema Selection.....	31
CA IDMS DMLO Session Mainline Screens .....	32
Record Display Screen .....	32
Area Display Screen .....	34
Data Manipulation Screen .....	35
Options Screen.....	37
PF Key Display Screen .....	38
EQUATE Display Screen .....	39
Database Key Display Screen .....	40
Sets Display Screen .....	41
Scratch Display Screen .....	42
Menu/Assist Mode Screen .....	43
Expanded Command (LRF) Screen .....	45
Interrupt Screen .....	46
CA IDMS DMLO Session Termination .....	46
PROFILE Save Screen .....	47
Using the CA IDMS DMLO CLIST Processor.....	47
What is a CLIST?.....	48
CLIST Creation .....	48
CLIST Selection/Maintenance .....	49
CLIST Execution.....	49
CLIST Execution Modes .....	51
CLIST Commands .....	52
CLIST Examples .....	53

## Chapter 3: Commands

**57**

Syntax Processor .....	58
Syntax Notation .....	58
CA IDMS DML Commands and Restrictions.....	59
CA IDMS DML Control Statements .....	60
CA IDMS DML Retrieval Statements .....	61
Using FIND/OBTAIN Commands .....	62
FIND/OBTAIN Special Considerations.....	65
CA IDMS DML Modification Statements .....	65
CA IDMS DML Save Statements .....	66
ACCEPT/RETURN Special Considerations.....	67
LRF Commands .....	67

---

Entering LRF Commands .....	68
CA IDMS DML SCRATCH/QUEUE Processing .....	69
Abbreviations .....	69
Scratch Display .....	69
Scratch/Queue Entity Variables .....	69
Default Buffer Areas .....	70
Default Data and Length Assignment .....	70
CA IDMS DML Command Restrictions .....	73
CA IDMS DMLO Commands .....	74
DISPLAY Command .....	75
EDIT Command .....	76
EQUATE Command .....	77
EXEC Command .....	78
HELP Command .....	79
INIT Command .....	80
LEARN Command .....	80
LIST Command .....	81
MOVE Command .....	81
PRINT Command .....	82
READYLIST Command .....	82
SAVE Command .....	83
SET AUTOBIND Command .....	83
SET CLIST Command .....	84
SET CMDDISPLAY Command .....	84
SET COBDISPLAY Command .....	85
SET DEFENTK Command .....	85
SET EXIT Command .....	85
SET HEX/NATIVE Command .....	86
SET INVCHAR Command .....	87
SET LOWERCASE Command .....	87
SET LRF Command .....	88
SET MAPIN Command .....	88
SET MENU Command .....	89
SHOW AREANAMES Command .....	89
SHOW EQUATES Command .....	89
SHOW KEYPADS Command .....	90
SHOW OPTIONS Command .....	90
SHOW PFKEYS Command .....	90
SHOW RECORDNAMES Command .....	90
SHOW SCRATCH Command .....	91
SHOW SETNAMES Command .....	91
SHOW VARIABLES Command .....	91

---

PROFILE Maintenance Commands .....	91
User PROFILE Maintenance Commands.....	92
System Administrator PROFILE Maintenance Commands .....	93
CLIST Maintenance Commands .....	95
User CLIST Maintenance Commands .....	95
System Administrator CLIST Maintenance Commands .....	96
CLIST Commands .....	98
CLIST Control Commands.....	98
CLIST Documentation Commands.....	99
CLIST Iteration Commands .....	100
CLIST Status Test Command .....	101
CLIST Conditional Commands .....	102
CLIST Data Manipulation Command .....	103
CLIST Interrupt Commands .....	104
CLIST Record Processing Commands .....	105
PF Key Processing.....	106
Variable Symbols.....	107
Subschema Variables.....	107
Macro Variables.....	108
Positional Variables.....	109
Selection Variables.....	110
CLIST Variables.....	110
Extended Command Processing.....	111
Creating Executable Commands .....	111
Commands Summary .....	114
Variable Symbol Restrictions.....	114
Command Processing Examples .....	115

## **Chapter 4: Operation** **119**

CA IDMS DMLO System Requirements .....	119
Terminal Type .....	119
Library Requirements .....	119
Security Requirements .....	119
Online Documentation Print Utility.....	120
Installation.....	121
User Exit Module.....	122

## **Chapter 5: Messages** **125**

Overview.....	125
Messages Generated by CA IDMS DMLO .....	125
Severity Codes .....	125



---

## Appendix A: CLIST Editing Commands

181

Overview.....	181
Editing Commands .....	181
Scroll Options.....	181
Primary Commands.....	182
Line Commands .....	182
Program Function Keys .....	182
Entering Commands .....	183
Scroll Options.....	184
Primary Commands.....	185
BOTTOM Command.....	185
BOUNDS Command.....	185
CANCEL Command .....	185
CAPS Command .....	186
CHANGE Command .....	186
COPY Command.....	187
CURSOR Command .....	188
DOWN Command.....	188
EDITOR-ID Command.....	188
ECHO Command .....	189
END Command.....	189
ENTER Command.....	189
EXCLUDE Command.....	190
FIND Command.....	190
FIRST Command.....	191
KEYS Command.....	191
LAST Command .....	191
LEFT Command .....	192
LOCATE Command .....	192
MEMORY Command .....	193
NULLS Command.....	193
PROFILE Command.....	194
RCHANGE Command .....	194
RESET Command .....	196
RESHOW Command .....	196
RFIND Command .....	196
RIGHT Command .....	197
SAVE Command .....	197
TABB Command.....	197
TABF Command .....	197
TABS Command .....	198

---

TIME Command .....	198
TOP Command .....	198
UP Command .....	199
Line Commands .....	199
Entering Line Commands .....	199
How to Use Line Commands .....	200
A (after) Command .....	200
B (before) Command .....	200
BNDS (bounds) Command .....	200
COLS (columns) Command .....	201
C (copy) Command .....	201
D (delete) Command .....	202
X (exclude) Command .....	202
I (insert) Command .....	203
M (move) Command .....	203
R (repeat) Command .....	204
TABS Command .....	204
Text Manipulation Line Commands .....	204

## **Appendix B: Batch CLIST Processing** **205**

Overview .....	205
Work Files .....	205
Batch CLIST Request Cards .....	207
Batch CLIST Upload File .....	208

## **Index** **211**

# Chapter 1: General Information

---

This guide provides the information needed to navigate smoothly through a database using CA IDMS DMLO. In addition, the many features that CA IDMS DMLO offers are documented to assist you.

This section contains the following topics:

[What Is CA IDMS DMLO?](#) (see page 11)

[How Can CA IDMS DMLO Improve Productivity?](#) (see page 12)

[Is CA IDMS DMLO Easy to Use?](#) (see page 12)

[CA IDMS DMLO Features](#) (see page 13)

## What Is CA IDMS DMLO?

CA IDMS DMLO is an interactive CA IDMS productivity tool which allows ad hoc navigation, retrieval, and update of CA IDMS/DB.

CA IDMS DMLO is a full-screen editor for database records that is as easy to use as a text editor. For database navigation, CA IDMS DMLO uses an interactive data manipulation language compatible with CA IDMS DMLC. CA IDMS DMLO accesses dictionary information to provide COBOL-like structured record displays with element values, and to allow updating the database with changed values.

Because CA IDMS DMLO uses CA IDMS DMLC commands, it is an excellent learning tool for database navigation techniques. Commands are executed interactively, and results are immediately available.

CA IDMS DMLO also provides an extensive set of control commands which allow you to manage the online session. CA IDMS/DMLO includes security and audit capabilities for protecting the database from unauthorized access.

## How Can CA IDMS DMLO Improve Productivity?

Uses for CA IDMS DMLO span all aspects of database activity, including application development, database design and analysis, performance evaluation, production support, and CA IDMS training.

CA IDMS DMLO allows you to do the following:

- Create OOAK records
- Create and maintain test database structures
- Check the results of program testing
- Locate and fix data integrity problems in production databases
- Validate navigation logic before coding
- Test LRF path definitions
- Analyze the I/O efficiency of access strategies.

It is even possible to perform complicated mass changes to the database—all without special programs.

## Is CA IDMS DMLO Easy to Use?

CA IDMS DMLO allows you to read and update database records without learning a new access language, and without writing special programs. Further, the data is presented in a familiar structured format.

Although CA IDMS DMLO supports standard CA IDMS DML syntax, it has a broad range of features and extensions which allow you to perform the above functions using shorthand notation, PF keys, macro commands, and repeatable series of commands.

CA IDMS DMLO is a powerful interactive query/update tool for the experienced CA IDMS technician.

CA IDMS DMLO also provides facilities by which the less experienced CA IDMS programmer can gain familiarity with database techniques and CA IDMS DML syntax in a more structured environment.

## CA IDMS DMLO Features

### Multiple/Distributed Database Support

CA IDMS DMLO fully supports multiple and distributed database and dictionary configurations described by the CA IDMS DBNAME/DBNODE and DICTNAME/DICTNODE parameters.

### Multiple Teleprocessing Environment Support

CA IDMS DMLO operates in the CA IDMS/DC, CICS/VS, TSO, and VM/ESA teleprocessing environments. Some functions may not be available in all environments.

### Support of CA IDMS DML Commands

CA IDMS DMLO supports all CA IDMS Data Manipulation Language commands associated with database navigation. These commands are implemented in their COBOL syntax (CA IDMS DMLC).

### Support of CA IDMS LRF Commands

CA IDMS DMLO supports access to both LR and mixed subschemas. All LRF commands are available. Support for the WHERE clause supports complex arithmetic and boolean expressions.

Because an LRF command can be quite lengthy, CA IDMS DMLO provides for a much expanded command area which can be requested by an option of the CA IDMS DMLO SET command.

CA IDMS DMLO provides an excellent vehicle for the database administrator to develop, debug, and evaluate LRF path definitions.

### Support of CA IDMS/DC SCRATCH/QUEUE Commands

CA IDMS DMLO provides a complete implementation of CA IDMS/DC scratch/queue processing, including all standard CA IDMS DMLC syntax, as well as powerful enhancements.

## Access to Dictionary Records

CA IDMS DMLO allows you to access record descriptions stored in the dictionary which are not part of the subschema being processed. These records may be used as work areas or scratch/queue data areas.

## Extensive Set of CA IDMS DMLO Commands

CA IDMS DMLO provides an extensive set of commands which allow you to optimize your online session. These commands supplement the CA IDMS/DMLC commands and allow you to manipulate data, control the data display, define abbreviations, use PF keys, and analyze the environment.

## Session PROFILES

At session termination CA IDMS DMLO provides an opportunity to preserve user-defined characteristics of a CA IDMS DMLO session, allowing convenient restoration of those parameters when another session is started. This set of session parameters is called a PROFILE and includes specifications of the following:

- Session Control Parameters
  - Subschema, schema, version
  - DBNAME, DBNODE, DICTNAME, DICTNODE
  - Display and processing options
  - User exit specifications
- READY modes for areas
- Abbreviations and Macro Commands (EQUATE settings)
- PF Key Assignments

Multiple PROFILES are allowed for each user. The concept of "global" PROFILES is also supported; i.e., the system administrator may define generically useful PROFILES which are available to all users. For efficiency, these PROFILES are stored in a small database structure established at product installation.

At session signon you may also request a selection list of available PROFILES. This same screen provides PROFILE maintenance capabilities, allowing you to COPY, RENAME, and KILL (delete) PROFILES.

## CLIST Programming

The CA IDMS DMLO CLIST facility provides a means for creating and executing sequences of CA IDMS/DML and CA IDMS DMLO commands. Up to nine variable arguments (%1-%9) may appear within the CLIST, and provide a means for defining general purpose routines. The CLIST processor includes structured constructs for IF-ELSE, looping, data manipulation, nesting of CLIST calls, and a prompt facility for arguments. CA IDMS DMLO includes an SPF-like text editor for entry and updates to the CLIST programs.

While the CA IDMS DMLO CLIST processor is not intended to be a full-featured interpretive programming language, it does offer adequate facilities to prepare utility routines of some complexity.

THE CLIST facility also provides for "global" CLISTs which may be created and maintained by the CA IDMS DMLO administrator but accessed by all users.

At any time during the session you may request a selection list of available CLISTs. You may also request COPY, RENAME, and KILL (delete) maintenance functions.

## Comprehensive Online Documentation

At any time during the CA IDMS DMLO session, you may press an installation-defined PF key to access CA IDMS DMLO online documentation. The online documentation is intended to provide extensive information at the terminal when you need it.

Online documentation includes:

- CA IDMS DMLO screen information
- CA IDMS DML command restrictions which apply in CA IDMS DMLO
- Comprehensive guide to CA IDMS DMLO commands
- Online message facility.

## Data Manipulation and Display

CA IDMS DMLO uses the dictionary to obtain record and element definitions necessary to allow a COBOL-format structured display of database and work records. CA IDMS DMLO allows you to select the format for display and entry of data into these fields. Possible formats are NATIVE (dictionary definition), HEX, and DUMP (simultaneous character and hex).

CA IDMS DMLO performs extensive data editing of data content and entry, and will detect and indicate invalid or non-displayable data. You may select AUTOHEX display mode, in which CA IDMS DMLO will automatically alter the display format to HEX for fields containing invalid data.

## PF Key Usage

CA IDMS DMLO allows you to associate frequently used commands with PF keys, thus reducing your keystrokes. You may assign commands to PF keys in two ways. First, you may enter an executable command and press a PF key rather than the ENTER key. This assigns the command to the PF key. Second, you may enter the "KEYS" command. This will display all current PF key settings, and allow updates to multiple keys, by simply entering the desired command opposite the PF key number. An additional capability of PF key processing is the ability to merge command line input with the command string represented by the PF key.

Any PF key assignments you make can be saved in a PROFILE.

## EQUATE Facility

Another time-saving facility of CA IDMS DMLO is the availability of a logical set of predefined abbreviations for CA IDMS DML keywords, and the ability for you to define additional abbreviations (EQUATES) for frequently used subschema entities and command phrases. These phrases may be partial or prototype macro commands, and may include previously equated words or phrases. Any EQUATES you define can be saved in a PROFILE.

## Selection Lists

CA IDMS DMLO provides "pop-up" selection lists for subschema entities. Such lists can eliminate the need for the user to recall the names of all records, sets, and areas in a given subschema, or active scratch ID's. These selection lists are invoked by the appearance of ?R, ?S, ?A or ?T within, or instead of, a command.

## Macros and Variables

CA IDMS DMLO provides for the use of variable symbols to construct executable CA IDMS DML commands. These symbols are of two types. First, there are the symbols representing active DB/DC entities &D, &R, &S, &A, &T, &Q. Second, there are the macro variables, &1 - &9, @1 - @9, which represent positional tokens from the command line to be merged with a prototype command.

For a complete description of the capabilities, of CA IDMS DMLO variable processing, see [CA IDMS DMLO Session](#) (see page 21).



## Access to CA IDMS Statistics

CA IDMS DMLO allows you to evaluate the efficiency of database access strategies by providing access to CA IDMS statistics. These statistics are always available as cumulative values since the start of the session. In addition, you may "reset" the values to allow determination of incremental statistics associated with execution of a smaller number of commands. Effectively, CA IDMS DMLO calculates the difference in cumulative statistics between two points that you specify.

## Extensive DBKEY Capabilities

CA IDMS DMLO provides extensive facilities for manipulation and display of database keys. Whenever a CA IDMS DMLC command requires a database key, you may use page-line, hexadecimal, or decimal literals. In addition, CA IDMS DMLO provides ten KEYPADS which may serve as source/target fields for commands. The SHOW KEYPADS command will display all three formats of these KEYPADS. Thus, you can use CA IDMS DMLO as a modest hex calculator.

In addition to providing data-base keys, page-information values may be supplied on the FIND/OBTAIN DML statement. CA IDMS DMLO provides ten page-info KEYPADS PGR0 - PGR9. The SHOW KEYPADS command will display both DBKEY and PAGE-INFO values.

## Menu/Assist Mode

The Menu/Assist Mode of CA IDMS DMLO can provide the less experienced CA IDMS user with a more structured environment which requires far less detailed knowledge of CA IDMS DML syntax. This mode can also assist in learning the full syntax while being productive with CA IDMS/DMLO.

CA IDMS DMLO provides the following features in Menu/Assist Mode:

- Structured, simplified presentation and entry of CA IDMS DMLC commands
- Instructional echo of standard syntax
- Installation-defined menu/assist display format
- Full access to all CA IDMS DMLO capabilities
- Ability to enter or exit Menu/Assist Mode using CA IDMS DMLO SET MENU or OPTIONS command

## Access Security

Three levels of security are available to CA IDMS DMLO users. You can select the degree of database protection that is needed at your installation.

- **Level One Security**—You may choose to bypass the CA IDMS DMLO security feature. If you decide not to use security, CA IDMS DMLO will verify only that the subschema name entered at signon is valid and usable.
- **Level Two Security**—Security is achieved through user ID and password entered at signon. CA IDMS DMLO allows you three chances to enter your correct password and ID.
- **Level Three Security**—At this security level CA IDMS DMLO verifies both password and ID (Level Two) and also makes certain that the user is authorized to access the subschema requested.

**Note:** Any access restrictions imposed by Level Three security will be reflected in the list of subschemas presented if a subschema selection list is requested at signon.

You can augment any CA IDMS DMLO level by also restricting access by ready mode. You can change the standard ready modes CA IDMS DMLO will process at a global level, and at the user ID level. For more information, see the appropriate *CA IDMS Installation and Maintenance Guide*.

## Installation Customization

A customization program USDTPARM distributed with CA IDMS DMLO allows the specification of CA IDMS DMLO parameters at installation. Note that some of these parameters are changeable by each user during a CA IDMS DMLO session (and those changes preserved in a PROFILE); while others are permanent in the sense that USDTPARM must be changed.

USDTPARM allows specification of the following session defaults:

- System administrator user IDs
- Signon screen options and default values
- SET command options
- PF key assignments
- Standard abbreviations
- Displayable characters—those which will not trigger an 'INVALID DATA' indication
- User exit specifications and messages
- Menu/Assist display format
- Permanently excluded subschemas, independently of normal CA IDMS DMLO security.

For more information, see the *CA IDMS Installation and Maintenance Guide—z/OS*.

## Mixed Page Group Support

CA IDMS DMLO allows a single CA IDMS rununit to access data from different page groups using the mixed page group support feature.

This feature also permits a single rununit to access a database with segments that have been defined with a different maximum for the number of records that can be stored on a page. You can specify a different maximum for each page group.

You can define up to 32,767 different page groups. A single page group can qualify up to 16 million pages which significantly increases the size of a database that can be accessed from a single rununit. Furthermore, by allowing portions of a database to have different maximum records per page, you can use the available pages more effectively.

Exit 34 is provided for use with the mixed page group support feature. You can use this exit to help identify and correct applications that may require modification to function correctly when the mixed page group support feature is enabled. For more information about user exits, see the *CA IDMS System Operations Guide*.

## Enter Key Usage

CA IDMS DMLO allows you to change the default setting for the Enter key. By changing the default, the last command on the DMLO command line is re-executed. This feature is useful, for example, when navigating a database to repeat FIND/OBTAIN NEXT/PRIOR records without having to overwrite data on the command line.

## User Exit Module

CA IDMS DMLO gives you the ability to customize an assembler user exit module which can be invoked after each DML command is executed. Upon return from the user exit to DMLO a return code value can be set to indicate an error condition and termination or continued processing.

**More information:**

[FIND/OBTAIN Special Considerations](#) (see page 65)

# Chapter 2: CA IDMS DMLO Session

---

This section contains the following topics:

[Overview](#) (see page 21)

[Introduction to the CA IDMS DMLO Session](#) (see page 24)

[CA IDMS DMLO Session](#) (see page 25)

[CA IDMS DMLO Session Mainline Screens](#) (see page 32)

[CA IDMS DMLO Session Termination](#) (see page 46)

[Using the CA IDMS DMLO CLIST Processor](#) (see page 47)

## Overview

This chapter is a step-by-step guide that shows you how to use CA IDMS DMLO. It includes a conceptual view of the CA IDMS DMLO session, gives information on initiating a session and executing commands, and describes how to interrupt and terminate a session. More detailed CA IDMS DMLO screen information is available through the online documentation.

## Text Split Command

**TS number-of-lines**

where:

**number-of-lines**-specifies the number of lines to be inserted between the split line. The default is 1.

The TS line command splits the text at the cursor so that you can insert text. The text following the cursor is moved to the left margin of the paragraph and an additional line is inserted.

See the online documentation for examples of the TS line command.

## Text Flow Command

**TF rb**

where:

**rb**-specifies the right bound for the text.

The TF command starts processing at the current line and flows text upward to the end of a paragraph. The end of a paragraph may be indicated by a:

- Blankline
- Change in indentation
- Special characters

Temporary lines such as COLS or BNDS are deleted before text is flowed.

A single blank separates existing text from the words that are flowed upward from a lower line. When the end of a sentence is detected, two blanks are inserted.

See the online documentation for examples of Text Flow parameters.

## Text Entry Command

**TE number-of-Lines**

where:

**number-of-lines**-specifies the number of blanklines requested.

The TE line command formats the screen with an unnumbered open text entry area which may be used without regard for line overflow. The cursor is positioned at the beginning of the first line and the remainder of the screen is blank. After you type the data and press the ENTER key, the text is flowed into a paragraph format.

If you type a number after the TE command, open the text entry area provided for only that number of lines.

See the online documentation for examples of TE command examples.

## Destructive Line Shift Examples

**number-of-columns**

where:

**number-of-columns** specifies the number of lines to shift. The default is 1.

**((number-of-columns))number-of-columns** **((number-of-columns). .((  
))number-of-columns. . .))**

Destructive line shift moves the text a specified number of columns to the right-) or left-(. When the shift causes text to exceed the bound position, that text is discarded.

See the online documentation for lineshift examples.

**number-of-columns**

Specifies the number of columns to shift. The default is 1.

**()**

Specifies that the line be shifted to the left or right one column.

**(n)n**

Specifies that the line be shifted to the left or right *n* columns.

**(( ))**

Specifies the first and last lines of a block to be shifted to the left or right one column.

**((n ))n**

Specifies the first line of a block to be shifted to the left or right *n* columns. Use a (( or )) to mark the last line of the block.

## Protective Line Shift Command

### **number-of-columns**

where:

### **number-of-columns**

Specifies the columns to shift. The default is 1.

**<[number-of-columns] >[number-of-columns] <<[number-of-columns]. . <<  
>>[number-of-columns]. . >>**

Protective line shift moves the text a specified number of columns to the right-> or left-<. When the shift causes the text to exceed the bound position, that text is retained and the shift operation is not completed.

See the online documentation for protective line shift examples.

### **number-of-columns**

Specifies the number of columns to shift. The default is 1.

**< >**

Specifies that the line be shifted to the left or right one column.

**<n >n**

Specifies that the line be shifted to the left or right *n* columns.

**<< >>**

Specifies the first and last lines of a block to be shifted to the left or right one column.

**<<n >>n**

Specifies the first line of a block to be shifted to the left or right *n* columns. Use a << or >> to mark the last line of the block.

## Introduction to the CA IDMS DMLO Session

A typical CA IDMS DMLO session resembles the logic of a CA IDMS batch program. The comparison shown below illustrates the similarities:

<b>CA IDMS DMLO Session Component</b>	<b>Batch Program Component</b>
SIGNON	BIND subschema
* BIND records	BIND records
* READY areas	READY areas



CA IDMS DMLO Session Component	Batch Program Component
CA IDMS DML and CA IDMS DMLO commands	Body of program
FINISH/ROLLBACK	FINISH/ROLLBACK
* PROFILE save	STOP RUN

\*Those components marked with "\*" are optional in a CA IDMS DMLO session. The functions are still essential parts of a CA IDMS DMLO session, but capabilities allow for automatic processing of these functions.

## CA IDMS DMLO Session

The detailed steps you will follow during a CA IDMS DMLO session will not follow a fully structured sequence. This is because CA IDMS DMLO allows you to determine not only what functions are appropriate for your processing needs, but also the sequence in which they are applied. However, the following list summarizes required and optional steps to be followed:

1. Invoke CA IDMS DMLO by entering the task code assigned at installation.
2. If quick-in startup is not used, complete the Signon screen.
  - From the Signon screen, you can request the Subschema List screen
  - From the Subschema List screen, you can select a subschema
  - From the Signon screen, you can request the PROFILE List screen
  - From the PROFILE Selection/Maintenance screen, you can perform PROFILE maintenance
  - From the PROFILE Selection/Maintenance screen, you can select a PROFILE
3. After startup is complete, CA IDMS DMLO may present the Record Display screen.
  - You may BIND records at this point, or defer the BIND
4. If the Record Display Screen is not presented, CA IDMS DMLO presents the Area Display screen showing READY mode of each area.
  - You may READY areas at this point, or defer the READY
5. From this point on, all screen displays are determined by your actions.
6. You may issue any CA IDMS DML command. These normally result in the display of the Data Manipulation screen.
  - You can alter the data content of that record.
  - You can issue CA IDMS DML modification commands for that record.

7. You may issue any CA IDMS DMLO command. Many of these commands result in a particular screen being displayed. Details are described below in the discussions of each screen format.
8. When you have completed all database activities, you may issue a ROLLBACK or FINISH command.
  - If you specify the SAVE or NOSAVE option, that option will be applied to the specified PROFILE, and the session will be terminated.
  - If neither SAVE nor NOSAVE is specified, the PROFILE Save screen is displayed.
9. You may specify which, if any, components of a PROFILE you wish to save, and the PROFILE name to be created or updated.
  - The session will be terminated after performing the requested PROFILE function.

## Invoking CA IDMS DMLO

There are two ways to begin a CA IDMS DMLO session. Standard session startup requires that you complete the Signon Screen before entering the session mainline. Quick-in session startup allows you to specify a PROFILE name at the time the CA IDMS DMLO task is invoked, and to bypass the Signon screen.

### Standard Session Startup

The standard method of session startup consists of the following steps:

1. Enter the CA IDMS DMLO task code. This results in display of the Signon screen.
2. Enter required information on the Signon screen.
3. If you enter an unknown PROFILE name or press the indicated PF key CA IDMS DMLO will display the PROFILE Selection/Maintenance screen. (See the following example).
4. If you press the indicated PF key, you can request online documentation for the Signon details.
5. If you enter a "?" in the subschema or schema field, CA IDMS DMLO displays a list of subschemas which you are authorized to access.
6. When you have provided sufficient Signon screen information CA IDMS DMLO will complete the session startup and proceed to the next screen.

7. If the AUTOBIND is not in effect and you did not specify a ready mode on the Signon screen, CA IDMS DMLO will present the Record Display screen.
  - You may BIND records in the list.
  - You may enter the BIND command now, or at any time during the session.
  - You may SET AUTOBIND ON to eliminate the necessity for doing any explicit BIND.
8. If AUTOBIND is in effect or you specified a ready mode on the Signon screen, CA IDMS DMLO will present the Area Display screen. This list will show the current ready mode, if any, of each area in the current subschema.
  - You may enter a ready mode next to any area(s) in the list.
  - You may issue an explicit CA IDMS DML READY command.
  - You may defer any further READY activity to a later point in the session.
9. You may now enter any CA IDMS DML or CA IDMS DMLO commands.

```

===== CA IDMS DML ONLINE RELEASE nn.nn =====
USER ID    => FULAL01                                PASSWORD =>
PROFILE NAME=>                                READY MODE=> (R=RETR/U=UPDT/P=PROF)
SUBSCHEMA  =>                                SCHEMA    =>          VERSION  =>
                                                DICTNAME  =>          DICTNODE =>
                                                DBNAME    =>          DBNODE   =>
                                                PRINT CLASS=> A      INTERRUPT=> PA1
                                                LOGICAL DISPLAY WIDTH=> 080      LOWER CASE=> N

PF1 /PF13 : HELP
PF5 /PF17 : PROFILE LIST

ALL RIGHTS RESERVED                                (C) 2007
===== CA, INC. =====

```

## Required Signon Parameters

The CA IDMS DMLO online documentation provides a detailed explanation of the contents of the Signon screen. However, it is important to note that you must provide a subschema name in order to begin a CA IDMS DMLO session. You may provide the subschema explicitly by entry on the Signon screen or selection from a list, or implicitly by requesting a PROFILE.

Entry of a schema name and version on the Signon screen is only necessary if the subschema name entered is not unique in the dictionary specified by DICTNAME/DICTNODE.

Whether you specify PROFILE or subschema on the Signon screen, you may specify READY mode.

**Note:** When you are communicating across CV's, you must specify DICTNAME and DICTNODE on the DMLO Signon Screen.

## Quick-In Session Startup (CA IDMS/DC only)

If you know the name of a PROFILE you wish to use, you may bypass the Signon screen as follows:

1. Enter the following command at "ENTER NEXT TASK CODE" :

```
DML0 profile-name ready-mode
```

Where:

**profile-name**

The PROFILE to be used for this CA IDMS DMLO session

**ready-mode**

Abbreviation for area READY specification (optional):

- U (shared Update)
- R (shared Retrieval)
- P (as previously stored in the PROFILE)

2. CA IDMS DMLO loads the specified PROFILE, readies all areas as indicated, and displays the Area Display screen.
  - You may enter a ready mode next to any area(s) in the list.
  - You may issue an explicit CA IDMS DML READY command.
  - You may defer any further READY activity to a later point in the session.
3. You may now enter any CA IDMS DML or CA IDMS DMLO commands.

If a profile name is unknown, the Profile Selection screen will be displayed. Thus, you may bypass the Signon screen and directly enter the Profile Selection screen by typing "DMLO ?" at the "ENTER NEXT TASK CODE" prompt.

## PROFILE Selection/Maintenance

The PROFILE Selection/Maintenance screen is displayed if you press the indicated PF key from the Signon screen. It will also be displayed if you request an unknown PROFILE name (in either standard or quick-in session startup).

You may take the following actions from this screen:

- You may select a PROFILE for use in this session by placing an "S" next to the item of interest.
- You may delete any non-global PROFILES (i.e., owner not = "SYS") by placing "K" next to the item(s) to be deleted.
- You may enter any PROFILE maintenance commands.
- You may enter the "LIST" command to refresh the list of PROFILES being displayed. Execution of maintenance commands will not result in an automatic refresh.

For more information on the commands that may be used at this screen, see [PROFILE Maintenance Commands](#) (see page 91).

ACTION	PROFILE	SRC	SUBSCH	SCHEMA	DBNAME	DICTNAME	USED	UPDT
-	EMPDEMO		EMPSS01	EMPSCHM	EMPDEMO		mmddy	mmddy
-	EMPPROF		EMPSS01	EMPSCHM			mmddy	mmddy
-	RALPH		IDMSNwKA	IDMSNTWK			mmddy	mmddy
-	TESTINVC		EMPSS01	EMPSCHM			mmddy	mmddy
-	X		EMPSS01	EMPSCHM			mmddy	mmddy
-	IDMSNwKA	SYS	IDMSNwKA	IDMSNTWK			mmddy	mmddy

  

DML/O Rnn.nn ===== CA, INC.  
RECORD= STATUS= DBKEY=000000000-0000 KEY0=000000000-0000  
I6011 ACTION = (K)ill or (S)elect, enter maint command or PA1 to exit  
SUBSCHEMA= SCHEMA= VER=0000 COL 001-080 LINE 0001 OF 0006

## Subschema Selection

The Subschema Selection screen is displayed if you enter at least one "?" in either the subschema or schema field of the Signon screen. The list presented contains only subschemas for which you have access. The list is further restricted by matching subschema and schema names to the "mask" patterns entered as subschema and/or schema name.

Mask patterns are processed as follows:

1. "?", "\*" (asterisk), or " " (space) in the pattern will match any character in the name.
2. Any other character in the pattern must exactly match the corresponding character in the name.

For example, "??SSRT " will match "IDMSSRT " and "ABCSSRTX".

The list is sorted by subschema name if "?" appears anywhere in the subschema pattern, otherwise the list is sorted by schema.

You can enter "S" next to an item to use that subschema for this CA IDMS DMLO session.

ACTION	SCHEMA	VERSION	SUBSCHEMA	ERROR
-	EMPSCHM	0100	EMPSSLR1	
-	EMPSCHM	0100	EMPSS01	
-	ESSSCHEM	0001	ESSSUB00	
-	IDMSNTWK	0001	IDMSNwKA	
-	IDMSSECQ	0001	IDMSSECQ	
-	IDMSSECS	0001	IDMSSECS	
-	IDMSSECU	0001	IDMSSECU	
-	SSKSHEM	0001	SSKSUB00	
-	USDSHEM	0001	USDSUB00	
-	USOSCHM2	0001	USOSUB02	

DML/O Rnn.nn ===== CA, INC.  
RECORD= STATUS= DBKEY=0000000000-0000 KEY0=0000000000-0000

I6015 ACTION = (S)elect or PA1 to exit  
SUBSCHEMA=? SCHEMA= VER=0000 COL 001-080 LINE 0001 OF 0010

## CA IDMS DMLO Session Mainline Screens

Descriptions of the following screens encountered in a CA IDMS DMLO session mainline are as follows:

- Record Display screen
- Area Display screen
- Data Manipulation screen
- Options screen
- PF Key Display screen
- EQUATE Display screen
- Database Key Display screen
- Sets Display screen
- Scratch Display screen
- Menu/Assist Mode screen
- Expanded Command (LRF) screen
- Interrupt screen

### Record Display Screen

The Record Display screen is displayed:

- After signon if AUTOBIND is not in effect and you did not specify a ready mode.
- If you enter a simple BIND command.
- If you enter the SHOW RECORDS command.
- The selection variable ?R appears as part of, or instead of, a command. Possible actions are indicated by the informational message.



For information on selection variable processing., see [CA IDMS DMLO Session](#) (see page 21).

ACTION	RECORD	AREA	BND
-	001 COVERAGE	INS-DEMO-REGION	
-	002 DENTAL-CLAIM	INS-DEMO-REGION	
-	003 DEPARTMENT	ORG-DEMO-REGION	
-	004 EMPLOYEE	EMP-DEMO-REGION	
-	005 EMPOSITION	EMP-DEMO-REGION	
-	006 EXPERTISE	EMP-DEMO-REGION	
-	007 HOSPITAL-CLAIM	INS-DEMO-REGION	
-	008 INSURANCE-PLAN	INS-DEMO-REGION	
-	009 JOB	ORG-DEMO-REGION	
-	010 NON-HOSP-CLAIM	INS-DEMO-REGION	
-	011 OFFICE	ORG-DEMO-REGION	
-	012 SKILL	ORG-DEMO-REGION	
-	013 STRUCTURE	EMP-DEMO-REGION	

DML/O Rnn.nn ===== CA, INC.  
RECORD= STATUS=0000 DBKEY=0000000255-0008 KEY0=0000000000-0000  
BIND  
I6005 ACTION = (B)ind, (I)nit  
SUBSCHEMA=EMPSS01 SCHEMA=EMPSCHM VER=0100 COL 001-080 LINE 0001 OF 0013

## Area Display Screen

The Area Display screen is displayed:

- After signon if AUTOBIND is in effect or you specified a ready mode.
- If you enter the SHOW AREA command.
- If the selection variable ?A appears as part of, or instead of, a command.

Possible actions are indicated by the informational message on the screen.

For information on selection variable processing, see [CA IDMS DMLO Session](#) (see page 21).

```
ACTION      AREA      MODE
--         --
--      001 EMP-DEMO-REGION
--      002 INS-DEMO-REGION
--      003 ORG-DEMO-REGION

DML/O Rnn.nn ===== CA, INC.
RECORD=          STATUS=0000 DEKEY=0000000255-0008 KEY0=0000000000-0000
I6006 ACTION = READY mode
SUBSCHEMA=EMPSS01  SCHEMA=EMPSCHM  VER=0100  COL 001-080  LINE 0001 OF 0003
```

## Data Manipulation Screen

After session startup, you may issue CA IDMS DML commands to access database records or scratch/queue records. Any command that results in data transfer will cause the appropriate record buffer to be formatted and displayed, that is, FIND will not alter the display, but OBTAIN and GET will.

You may alter data values on this screen and issue modification commands.

The presentation format of data may be controlled in some detail with CA IDMS DMLO options and commands. For detailed information on the SET and DISPLAY commands, see [CA IDMS DMLO Session](#) (see page 21).

```

mm/dd/yy.....RECORD : EMPLOYEE.....hh:mm:ss
02 EMP-ID-0415.....N 0120
02 EMP-NAME-0415.....G
03 EMP-FIRST-NAME-0415.....A MICHAEL
03 EMP-LAST-NAME-0415.....A ANGELO
02 EMP-ADDRESS-0415.....G
03 EMP-STREET-0415.....A 507 CISTINE DR
03 EMP-CITY-0415.....A WELLESLEY
03 EMP-STATE-0415.....A MA
03 EMP-ZIP-0415.....G
04 EMP-ZIP-FIRST-FIVE-0415.....A 01568
04 EMP-ZIP-LAST-FOUR-0415.....AX 40404040
02 EMP-PHONE-0415.....N 6178870235
02 STATUS-0415.....A* FFFF
02 SS-NUMBER-0415.....N 127675593
02 START-DATE-0415.....G
03 START-YEAR-0415.....N 79
03 START-MONTH-0415.....N 09
03 START-DAY-0415.....N 08
=====
DML/O Rnn.nn ===== CA, INC.
RECORD=EMPLOYEE STATUS=0000 DBKEY=0000075083-0001 KEY0=0000000000-0000
DIS EMPLOYEE
SUBSCHEMA=EMPSS01 SCHEMA=EMPSCHM VER=0100 COL 001-080 LINE 0001 OF 0027

```

```

mm/dd/yy.....RECORD : MPLYOEE.....hh:mm:ss
02 EMP-ID-0415.....NX F0F1F2F0
02 EMP-NAME-0415.....AX D4C9C3C8C1C5D3404040C1D5C7C5D3D6
=====+0016.....AX 404040404040404040
03 EMP-FIRST-NAME-0415.....AX D4C9C3C8C1C5D3404040
03 EMP-LAST-NAME-0415.....AX C1D5C7C5D3D64040404040404040
02 EMP-ADDRESS-0415.....AX F5F0F740C3C9E2E3C9D5C540C4D94040
=====+0016.....AX 40404040E6C5D3D3C5E2D3C5E8404040
=====+0032.....AX 404040D4C1F0F1F5F6F840404040
03 EMP-STREET-0415.....AX F5F0F740C3C9E2E3C9D5C540C4D94040
=====+0016.....AX 40404040
03 EMP-CITY-0415.....AX E6C5D3D3C5E2D3C5E8404040404040
03 EMP-STATE-0415.....AX D4C1
03 EMP-ZIP-0415.....AX F0F1F5F6F840404040
04 EMP-ZIP-FIRST-FIVE-0415.....AX F0F1F5F6F8
04 EMP-ZIP-LAST-FOUR-0415.....AX 40404040
02 EMP-PHONE-0415.....NX F6F1F7F8F8F7F0F2F3F5
02 STATUS-0415.....AX FFFF
02 SS-NUMBER-0415.....NX F1F2F7F6F7F5F5F9F3
DML/O Rnn.nn ===== CA, INC.
RECORD=EMPLOYEE STATUS=0000 DBKEY=0000075083-0001 KEY0=0000000000-0000
DIS EMPLOYEE HEX
SUBSCHEMA=EMPSS01 SCHEMA=EMPSCHM VER=0100 COL 001-080 LINE 0001 OF 0031
    
```

Preceding the data portion of the Data Manipulation screen there is a 2-character indicator of each field's data format. The first position describes the storage mode of the field. The second position indicates if the data is being displayed in its native mode (" "), is being displayed in hex based on the SET command ("X"), or is being displayed in hex because the AUTOHEX option is effective and the field contains invalid data for its usage mode ("\*"). AUTOHEX display formats are also highlighted.

```

mm/dd/yy.....RECORD : EMPLOYEE.....hh:mm:ss
0.1.2.3. 4.5.6.7. 8.9.A.B. C.D.E.F. 0...4...8...C...
0000 F0F1F2F0 D4C9C3C8 C1C5D340 4040C1D5 => 0120MICHAEL AN <=
0016 C7C5D3D6 40404040 40404040 40F5F0F7 => GELO 507 <=
0032 40C3C9E2 E3C9D5C5 40C4D940 40404040 => CISTINE DR <=
0048 40E6C5D3 D3C5E2D3 C5E84040 40404040 => WELLESLEY <=
0064 D4C1F0F1 F5F6F840 404040F6 F1F7F8F8 => MA01568 61788 <=
0080 F7F0F2F3 F5FFFFF1 F2F7F6F7 F5F5F9F3 => 70235 127675593 <=
0096 F7F9F0F9 F0F8F0F0 F0F0F0F0 F5F7F0F4 => 7909080000005704 <=
0112 F0F50000 => 05_ <=

DML/O Rnn.nn ===== CA, INC.
RECORD=EMPLOYEE STATUS=0000 DBKEY=0000075083-0001 KEY0=0000000000-0000
DIS EMPLOYEE DUMP
SUBSCHEMA=EMPSS01 SCHEMA=EMPSCHM VER=0100 COL 001-080 LINE 0001 OF 0008
    
```

## Options Screen

The Options screen is displayed when you enter the SHOW OPTIONS command. This screen displays information from the Signon screen, as well as a series of options controlled by the SET command. Any or all of these options can be changed at once. Make any desired changes, and press the ENTER key to return to the prior display.

For detailed information on the SET command options., see [CA IDMS DMLO Session](#) (see page 21) The specific forms of the SET command which correspond to the entries on the Options Screen are as follows:

1. SET MENU
2. SET COBDISPLAY
3. SET AUTOHEX
4. SET CMDDISPLAY
5. SET LRF
6. SET AUTOBIND
7. SET MAPIN
8. SET CLIST
9. SET EXIT
10. SET INVCHAR

```

===== CA IDMS DML ONLINE RELEASE nn.nn =====
USER ID    => FULAL01
PROFILE NAME=> EMPPROF      READY MODE=>
SUBSCHEMA  => EMPSS01      SCHEMA   => EMPSCHM   VERSION  => 0100
                                           DICTNAME =>
                                           DBNAME   =>
                                           PRINT CLASS=> A      INTERRUPT=> PA1
                                           LOGICAL DISPLAY WIDTH=> 080      LOWER CASE=> N

1. DML/O "MENU" MODE   => N      6. AUTO BIND RECORDS  => Y
2. COBOL DISPLAY FORMAT=> Y      7. FAST DATA INPUT MODE=> Y
3. AUTOHEX DATA DISPLAY=> Y     8. FAST CLIST EXEC MODE=> Y
4. AS USED CMND DISPLAY=> Y     9. ACTIVE USER EXIT   => N
5. LRF EXPAND CMND AREA=> N     10. INVALID CHAR REPLACE=> _

ALL RIGHTS RESERVED                                     (C) 2007
===== CA, INC. =====

```

## PF Key Display Screen

The PF Key Display screen appears when you enter the SHOW KEYS command. You may view settings of all PF/PA keys by scrolling up/down. You may also enter or change any PF key text assignment by keying a command or phrase next to any PF key label.

**Note:** A PF key may also be assigned by pressing that PF key rather than the ENTER key to execute a command. For more information, see [Extended Command Processing](#) (see page 111).

```
mm/dd/yy.....PF KEY ASSIGNMENTS.....hh:mm:ss
PA1 : INTERRUPT KEY
PA2 : FINISH NOSAVE
PA3 :
PF1 : HELP
PF2 : KEYS
PF3 : END
PF4 : DISPLAY &D
PF5 :
PF6 :
PF7 : UP
PF8 : DOWN
PF9 :
PF10:
PF11:
PF12:
DML/O Rnn.nn ===== CA, INC.
RECORD=                STATUS=0000  DBKEY=000000255-0008  KEY0=000000000-0000
KEYS
  SUBSCHEMA=EMPSS01  SCHEMA=EMPSCHM  VER=0100  COL 001-080  LINE 0001 OF 0030
```

## EQUATE Display Screen

The EQUATE Display screen appears when you enter the SHOW EQUATES command. This display is a list of all abbreviations created during this session or loaded as part of a PROFILE. Standard abbreviations are not displayed.

This screen has no provisions for data update. Changes are made only with the EQUATE command. See [CA IDMS DMLO Session](#) (see page 21) for detailed information on the EQUATE command and its use in Extended Command Processing.

```
mm/dd/yy.....USER-DEFINED EQUATES.....hh:mm:ss
EMPID..... EMP-ID-0415
EMPLNAME... EMP-LAST-NAME-0415
EMPFNAME... EMP-FIRST-NAME-0415
EMPNAME.... EMP-NAME-0415

DML/O Rnn.nn ===== CA, INC.
RECORD=                STATUS=0000  DEKEY=0000000255-0008  KEY0=0000000000-0000
SHOW EQUATES
SUBSCHEMA=EMPS01  SCHEMA=EMPSCHM  VER=0100  COL 001-080  LINE 0001 OF 0004
```

## Database Key Display Screen

The Database Key Display screen appears when you enter the SHOW KEYPADS command. It displays database keys and page-information which are part of the subschema-control-block as well as the ten CA IDMS DMLO database work areas for DBKEYS (KEY0 - KEY9) and PAGE-INFO values (PGR0 - PGR9). Note that three equivalent formats are displayed.

This screen has no provision for data update. Changes to DBKEY and PAGE-INFO values occur as a result of executing CA IDMS DML commands, or in the case of DBKEYS as a result of MOVE commands which specify one of the database key fields as a target.

```

mm/dd/yy.....DATABASE KEY FIELDS.....hh:mm:ss
SSCTRL: DBKEY.....          75008-0001   X'00928001'   F'9601025'
SSCTRL: DIRECT-DBKEY.
SSCTRL: PGINFO              1001-0000   X'03E90007'   F'65601543'
KEYPAD: KEY0.....
KEYPAD: KEY1.....
KEYPAD: KEY2.....
KEYPAD: KEY3.....
KEYPAD: KEY4.....
KEYPAD: KEY5.....
KEYPAD: KEY6.....
KEYPAD: KEY7.....
KEYPAD: KEY8.....
KEYPAD: KEY9.....
KEYPAD: PGR0.....          1001-0000   X'03E90007'   F'65601543'
KEYPAD: PGR1.....
KEYPAD: PGR2.....          1001-0000   X'03E90007'   F'65601543'
KEYPAD: PGR3.....
KEYPAD: PGR4.....
KEYPAD: PGR5.....
KEYPAD: PGR6.....
KEYPAD: PGR7.....
KEYPAD: PGR8.....
KEYPAD: PGR9.....
DML/O Rnn.nn ===== CA, INC.
RECORD=EMPLOYEE          STATUS=0000  DBKEY=0000075008-0001 KEY0=0000000000000000
SHOW KEYPADS

SUBSCHEMA=EMPXSS01  SCHEMA=EMPXSCHM  VER=0100  COL 001-080  LINE 0001 OF 0024
    
```



## Sets Display Screen

This screen is displayed if:

- You enter the SHOW SETS command
- You use the selection variable ?S as part of, or instead of, a CA IDMS DML or CA IDMS DMLO command.

If ?S is part of a command, you may select a setname by entering "S" next to the item of interest. Otherwise, the display is informational only.

For more information about selection variables, see [CA IDMS DMLO Session](#) (see page 21).

ACTION	SET	OWNER	MEMBERS	
-	001 COVERAGE-CLAIMS	COVERAGE	DENTAL-CLAIM	HOSPITAL-CLAIM
-			NON-HOSP-CLAIM	
-	003 DEPT-EMPLOYEE	DEPARTMENT	EMPLOYEE	
-	004 EMP-COVERAGE	EMPLOYEE	COVERAGE	
-	005 EMP-EMPOSITION	EMPLOYEE	EMPOSITION	
-	006 EMP-EXPERTISE	EMPLOYEE	EXPERTISE	
-	007 EMP-NAME-NDX	SYSTEM	EMPLOYEE	
-	008 JOB-EMPOSITION	JOB	EMPOSITION	
-	009 JOB-TITLE-NDX	SYSTEM	JOB	
-	010 MANAGES	EMPLOYEE	STRUCTURE	
-	011 OFFICE-EMPLOYEE	OFFICE	EMPLOYEE	
-	012 REPORTS-TO	EMPLOYEE	STRUCTURE	
-	013 SKILL-EXPERTISE	SKILL	EXPERTISE	
-	014 SKILL-NAME-NDX	SYSTEM	SKILL	

DML/O Rnn.nn ===== CA, INC.  
RECORD= STATUS=0000 DEKEY=0000000255-0008 KEY0=0000000000-0000  
OBT FIR IN ?S

## Scratch Display Screen

This screen is displayed if:

- You enter the SHOW SCRATCH command.
- You use the selection variable ?T as part of, or instead of, a CA IDMS DML or CA IDMS DMLO command.

You may delete a SCRATCH area by entering "K" next to the item(s) to be deleted.

If ?T is part of a command, you may select a scratch area by entering "S" next to the item of interest. Otherwise, the display is informational only.

For detailed information on selection variables, see [CA IDMS DMLO Session](#) (see page 21).

```

SUBSCHEMA=EMPSS01  SCHEMA=EMPSCHM  VER=0100  COL 001-080  LINE 0001 OF 0003
*===== SCRATCH AREA ID =====*
ACTION *----- CHAR -----* *----- HEX -----*
-      ALFTEST1          C1D3C6E3C5E2E3F14040404040404040
-      ALFTEST2          C1D3C6E3C5E2E3F24040404040404040
-      ALFTEST3          C1D3C6E3C5E2E3F34040404040404040

DML/O Rnn.nn ===== CA, INC.
RECORD=EMPLOYEE      STATUS=0000  DBKEY=0000075008-0001  KEY4=0000075007-0001
GET SCR ?T FIRST
ISQ01 ACTION = (K)ill, (S)elect -or- PA1 to exit
SUBSCHEMA=EMPSS01  SCHEMA=EMPSCHM  VER=0100  COL 001-080  LINE 0001 OF 0003
    
```

## Menu/Assist Mode Screen

The Menu/Assist Mode of CA IDMS DMLO provides a more structured environment which can relieve the less experienced CA IDMS user of needing detailed knowledge of CA IDMS DML syntax.

The initial Menu/Assist Mode screen is displayed when you enter the SET MENU ON command, or when you specify this option on the OPTIONS screen. This screen contains installation-defined instructional messages in the data display area.

After entry of the first command in Menu Mode, the upper portion of the screen becomes a data display and entry area.

You may switch back and forth between Menu/Assist Mode and "Expert" Mode as desired using the SET MENU OFF/ON command or the Options screen (by using the SHOW OPTIONS command).

```

You have selected the "MENU/ASSIST" mode of DML/O.
Please note the following :
>> All DML and DML/O commands will continue
to be recognized
>> Shorthand commands are now possible.
Each command consists of a code (formed
by concatenating parenthesized letters) and one or subschema
entities.
>> Notation for subschema entities is as follows:
R = record S = set A = area K = dbkey F = sortfield
<R> => record is optional S/A => either set or area is
required
>> If a name is not known, entering ?R, ?S, ?A will
display a selection list.
>> Any abbreviations entered via EQUATE commands
commands will be honored.
>> Entity variables &R, &S, &A will
be honored.
>> For example, the DML command OBTAIN FIRST EMPLOYEE WITHIN EMPLOYEE-AREA
could be entered as OF EMPLOYEE EMPLOYEE-AREA -or- OF EMPLOYEE &A

DML/O Rnn.nn ===== ERROR STATUS : 0000 ===== CA, INC.
*----- (O)BTAIN / (F)IND -----*
(C)ALC R (F)IRST <R> S/A (E)RASE (A)LL R (FIN)ISH
(D)BKEY <R> K (N)EXT <R> S/A (CO)NNECT R S (COM)MIT
(DU)P R (P)RIOR <R> S/A (DI)SCONNECT R S (ROL)LBACK
(CU)RRENT R/S/A (L)AST <R> S/A (M)ODIFY R
(O)WNER S (U)SING R S F (S)TORE R PF1: HELP
..COMMAND INPUT..
..MESSAGE OUTPUT.. DBKEY: 255-8

```

```

mm/dd/yy.....RECORD : EMPLOYEE.....hh:mm:ss
02 EMP-ID-0415.....N 0023
02 EMP-NAME-0415.....G
03 EMP-FIRST-NAME-0415.....A KATHERINE
03 EMP-LAST-NAME-0415.....A O'HEARN
02 EMP-ADDRESS-0415.....G
03 EMP-STREET-0415.....A 12 EAST SPEEN ST
03 EMP-CITY-0415.....A NATICK
03 EMP-STATE-0415.....A MA
03 EMP-ZIP-0415.....G
04 EMP-ZIP-FIRST-FIVE-0415.....A 02364
04 EMP-ZIP-LAST-FOUR-0415.....A 9999
02 EMP-PHONE-0415.....N 6178897134
02 STATUS-0415.....A 01
                                OBTAIN FIRST EMPLOYEE WITHIN &A
DML/O Rnn.nn ===== ERROR STATUS : 0000 ===== CA, INC.
*----- (OB)TAIN / FIND -----*
(C)ALC R (F)IRST <R> S/A (E)RASE (A)LL R (FIN)ISH
(D)BKEY <R> K (N)EXT <R> S/A (CO)NNECT R S (COM)MIT
(DU)P R (P)RIOR <R> S/A (DI)SCONNECT R S (ROL)LBACK
(CU)RRENT R/S/A (L)AST <R> S/A (M)ODIFY R
(O)WNER S (U)SING R S F (S)TORE R PF1: HELP
OBTAIN FIRST EMPLOYEE WITHIN &A
-OK- DBKEY: 75007-1
    
```

## Expanded Command (LRF) Screen

The standard Data Manipulation screen contains only one line for entry of CA IDMS DML and CA IDMS DMLO commands. This is normally sufficient. However, CA IDMS DML does allow for the possibility of very long commands. This could occur particularly in the case of complex WHERE expressions for LRF records. The Expanded Command screen is displayed when you enter the command SET LRF ON, or when you specify that option on the Options screen. This screen format allows for a multi-line command input area. If this is still insufficient for a particular command, you may use the EQUATE facility to establish abbreviations for phrases/words in the command.

You may switch back and forth between expanded command and standard format via the SET LRF OFF/ON command or the OPTIONS screen as desired.

```

mm/dd/yy.....RECORD : EMPLOYEE.....hh:mm:ss
02 EMP-ID-0415.....N 0023
02 EMP-NAME-0415.....G
03 EMP-FIRST-NAME-0415.....A KATHERINE
03 EMP-LAST-NAME-0415.....A O'HEARN
02 EMP-ADDRESS-0415.....G
03 EMP-STREET-0415.....A 12 EAST SPEEN ST
03 EMP-CITY-0415.....A NATICK
03 EMP-STATE-0415.....A MA
03 EMP-ZIP-0415.....G
04 EMP-ZIP-FIRST-FIVE-0415.....A 02364
04 EMP-ZIP-LAST-FOUR-0415.....A 9999
02 EMP-PHONE-0415.....N 6178897134
02 STATUS-0415.....A 01
02 SS-NUMBER-0415.....N 019556712
02 START-DATE-0415.....G
DML/O Rnn.nn ===== CA, INC.
OBT FIR EMPLOYEE IN &A

-OK-
STATUS=0000
LINE 0001 OF 0027

```

## Interrupt Screen

The Interrupt screen is displayed when you press the "interrupt" key during the mainline CA IDMS DMLO session. The interrupt key is a key pre-defined at CA IDMS DMLO installation and changeable for each session.

The Interrupt screen prompts you to enter an indicator of your "intentions"; i.e., to CONTINUE the session or to ROLLBACK/CONTINUE and terminate the session.

The screen exists for compatibility with earlier versions as well as protection against inadvertent use of the interrupt key.

**Note:** The interrupt key is also used to escape from selection lists which expect a selection to complete the syntax of a command. In such a case, the Interrupt screen will not be displayed.

```
===== CA IDMS DML ONLINE RELEASE nn.nn =====  
  
SELECT ONE OF THE FOLLOWING CHOICES  
  
FINISH  
ROLLBACK  
CONTINUE  
  
ALL RIGHTS RESERVED (C) 2007  
===== CA, INC. =====
```

## CA IDMS DMLO Session Termination

You may terminate the CA IDMS DMLO session by entering the FINISH command, or the ROLLBACK command without the CONTINUE option. If you do not specify a SAVE/NOSAVE option, the PROFILE Save screen is displayed. If you did specify one of those options, the requested PROFILE is performed and CA IDMS DMLO terminates. For more information about the FINISH and ROLLBACK commands, see [CA IDMS DMLO Session](#) (see page 21).

## PROFILE Save Screen

The PROFILE Save screen is displayed based on the session termination request as described above. This screen allows you to update the active PROFILE, or to create a new PROFILE. If a PROFILE was loaded at session startup, its name will appear on this screen. You may change the PROFILE name to create/update another. You may also specify which PROFILE components are to be updated. Only those items with a "Y" will be used for update.

```
===== CA IDMS DML ONLINE RELEASE nn.nn =====
SAVE SESSION PARAMETERS AS      ==> EMPPROF
SELECT/EXCLUDE (Y/N) ITEMS TO BE SAVED :
      SIGNON SCREEN VALUES ==> Y
      BIND/READY OPTIONS      ==> Y
      PF KEY SETTINGS         ==> Y
      EQUATE SETTINGS         ==> Y
AND PRESS ENTER TO SAVE      OR   PRESS PA1 TO BYPASS FUNCTION

I5012 ERROR-STATUS 0000 RETURNED FROM FINISH

ALL RIGHTS RESERVED                                (C) 2007
===== CA, INC. =====
```

## Using the CA IDMS DMLO CLIST Processor

The CA IDMS DMLO CLIST processor provides the facility for creating, maintaining, and executing sequences of CA IDMS DML and/or CA IDMS DMLO commands. Storage of CLIST modules and module-level maintenance is similar to storage and maintenance of PROFILES.

## What is a CLIST?

A CA IDMS DMLO CLIST is a sequence of CA IDMS DML and CA IDMS DMLO commands which may be executed as a mini-program. Any CA IDMS DML and CA IDMS DMLO command may be incorporated in a CLIST. In addition, there are special CLIST commands and constructs which allow a CA IDMS DMLO CLIST to incorporate looping and conditional capabilities.

A CLIST may be written using variable arguments which are supplied when execution of the CLIST is requested. This allows you to write general-purpose routines for frequently needed database functions.

Global CLISTs are supported. These CLISTs are prepared by the CA IDMS DMLO system administrator, and are accessible in read-only form to all users of the system. When you request a list of available CLISTs, global CLISTs are flagged "SYS".

## CLIST Creation

Use the EDIT command to create or update a CLIST. This command invokes the CA editor for CLIST text processing. This command is available only in a CA IDMS/DC environment.

**EDIT [S.]clist-name**

Where:

- **S.**—forces access to global CLIST if both a user and global CLIST of same name exist.
- **clist-name**—1-8 character CLIST identifier.

Details of the CA IDMS DMLO CLIST editing commands are discussed in [CLIST Editing Commands](#) (see page 181).

If you are not the CA IDMS DMLO system administrator, an attempt to edit a global CLIST forces you into a BROWSE rather than an EDIT session. You may, however, copy global CLISTs into one of your own CLISTs during an EDIT session.

```

EDIT — EDIT CLIST : GETCALC                                COLUMNS 001 072
COMMAND ==>                                                SCROLL ==> PAGE
***** ** TOP OF DATA ***** CA IDMS/DMLO **
000001 * GENERIC OBTAIN CALC ROUTINE
000002 :%1 = RECORD TO BE OBTAINED
000003 :%2 = CALC-KEY FIELD NAME
000004 :%3 = VALUE OF CALC-KEY TO BE READ
000005 *
000006 CINIT %1
000007 MOVE %3 TO %2
000008 OBTAIN CALC %1
000009 ENDC
***** ** BOTTOM OF DATA ***** CA IDMS/DMLO

```



## CLIST Selection/Maintenance

At any time during the session you may enter the LIST command. This will result in a selection list of available CLIST modules. The list includes CLISTs stored under your user ID as well as global CLISTs. If the first statement in a CLIST is a comment ("\*" in column 1), part of that comment will be displayed in the list.

You may enter module-level maintenance commands for your CLISTs. These commands allow you to COPY (possibly from another user), RENAME, and KILL (delete) CLISTs. For complete descriptions of the maintenance commands., see [CA IDMS DMLO Session](#) (see page 21).

You may enter CLIST maintenance commands at anytime during a session, not only when the CLIST Selection/Maintenance screen is displayed.

ACTION	CLIST	SRC	DESCRIPTION	USED	ADDED
-	PUTTESTQ		* BUILD TEST QUEUES	mmddy	mmddy
-	PUTTESTS		* BUILD TEST SCRATCH	mmddy	mmddy
-	TEST01		* NULL TEST CLIST	mmddy	mmddy
-	TEST02		* NULL TEST CLIST	mmddy	mmddy
-	TEST03		* NULL TEST CLIST	mmddy	mmddy
-	OBTCALC	SYS	* GENERIC OBTAIN CALC	mmddy	mmddy

```
DML/O Rnn.nn ===== CA, INC.
RECORD=employee      STATUS=0000  DBKEY=0000075008-0001 KEY4=0000075007-0001
LIST
I6010 ACTION = (E)dit, e(X)ecute, or (K)ill
```

## CLIST Execution

Use the following command to request execution of a CLIST:

```
EXEC [S.]clist-name a1 a2 a3 a4 a5 a6 a7 a8 a9
```

Where:

- **S.** - forces access to global CLIST if both a user and global CLIST of same name exist.
- **clist-name** - 1-8 character CLIST identifier.
- **a1 ... a9** - represent optional arguments for replacement of CLIST variables appearing within the body of the CLIST.

You may incorporate CLIST variables within any CLIST statements. These variables are of the form %1 ... %9. This variable capability allows you to prepare general-purpose modules for frequently used functions.

Arguments supplied with the EXEC statement may be record, area, set, scratch/queue names, element names, keywords, or literals of any acceptable format. The arguments will replace all occurrences of the corresponding variables within the CLIST statements at the time the statement is executed.

You may indicate that an optional argument is null by entering "," for that argument. This will cause the corresponding symbol to be replaced by a null string in the CLIST command. You may test if argument *n* is null with a conditional test for %NULL*n*. This conditional name is "true" if the corresponding argument is supplied as a null value.

The CLIST facility allows you to use DEFINE symbols (:%*n*) for documenting the use of CLIST arguments ahead of the executable CLIST commands. Not only will this provide documentation for a user who is editing the CLIST, but it allows the CLIST processor to prompt for missing arguments. If a CLIST command is encountered which includes a CLIST variable for which a corresponding argument has not been provided, the processor will display a screen which allows you to supply the missing argument value. If a DEFINE exists for the missing argument, the associated documentation is also displayed as shown below.

For more information, see [CLIST Editing Commands](#) (see page 181).

```
mm/dd/yy.....GETCALC  COMMAND PROMPT.....hh:mm:ss
THE FOLLOWING COMMAND CONTAINS UNRESOLVED CLIST ARGUMENTS. YOU MUST PROVIDE
VALUE(S) TO CONTINUE EXECUTION. AFTER ENTERING VALUES, PRESS THE "END" KEY

CINIT %1

:%1 = RECORD TO BE OBTAINED
:%1 001-064..
:%1 065-128..
:%1 129-192..
:%1 193-256..

DML/O Rnn.nn == CLIST = GETCALC          DEPTH C=01 R=00 ===== CA, INC.
RECORD=EMPLOYEE          STATUS=0000  DBKEY=0000075008-0001 KEY0=0000000000-0000
EXEC GETCALC

SUBSCHEMA=EMPSS01  SCHEMA=EMPSCHM  VER=0100  COL 001-080  LINE 0001 OF 0012
```

If you have no idea of what arguments are needed for a given CLIST, "EXEC clist-name?" will cause the CLIST processor to prompt for all arguments, and display all DEFINE information. Such documentation for any non-trivial CLIST is highly recommended. Note that a CLIST may EXEC another CLIST. Thus, you may define CLIST subroutines called by other CLISTS. CA IDMS DMLO will ensure that recursion does not occur; i.e., a CLIST does not appear at multiple levels in an active EXEC nest.

## CLIST Execution Modes

CA IDMS DMLO provides the following CLIST execution modes:

- FAST Mode
- STEP Mode
- STEP/LEARN Mode

Selection of execution mode is done with the SET CLIST FAST/STEP command or with the Options screen.

Specification of LEARN Mode is done with the LEARN ON/OFF command or with the Options screen.

### FAST Mode Execution

FAST Mode execution executes series of commands without interruption until a stopping point is reached. A stopping point is any error condition, a PAUSE or PROMPT command, or end of the CLIST. If an error condition occurs, execution will revert to STEP Mode. FAST Mode does not provide any screen displays until a stopping point occurs.

### STEP Mode Execution

STEP Mode execution executes one command at a time and displays the results of that command. You must press the ENTER key to cause the next command to be executed. In between CLIST commands you may enter additional commands to be executed. When you press the ENTER key without a command line entry, the CLIST will execute its next command.

During STEP Mode, CA IDMS DMLO displays the next command to be executed in the message area of the display.

### STEP/LEARN Mode Execution and Command Input

The LEARN Mode for command input during STEP Mode CLIST execution will execute and permanently insert into the CLIST source any commands which you enter when the CLIST is paused.

IFC and REPEAT commands may not be entered during LEARN Mode execution.

Use the LEARN ON/OFF command to initiate/terminate LEARN Mode. You may enter the command before starting CLIST execution, or at any time during execution. It may also appear as a command within the CLIST itself.

The CA IDMS DMLO display indicates that you are in LEARN/STEP Mode.

## CLIST Commands

CA IDMS DMLO's CLIST processor provides extensive CLIST capabilities. Detailed information on CLIST command syntax can be found in the online documentation and in [CA IDMS DMLO Session](#) (see page 21).

- CLIST Control Commands
  - EXEC
  - EXITC
  - ENDC
  - QUIT
- CLIST Iteration Commands
  - REPEAT
  - EXITR
  - ENDR
- CLIST Status Test Command
  - ON
- CLIST Conditional Commands
  - IFC
  - ENDIF
  - ELSE
- CLIST Data Manipulation Command
  - MOVE
- CLIST Interrupt Commands
  - PAUSE
  - RESUME
  - PROMPT
- CLIST Record Processing Commands
  - CBIND
  - CINIT

## CLIST Examples

This section contains three CLIST examples which use a variety of CA IDMS DMLO CLIST processing features:

**Note:** For detailed CLIST command syntax information, see the chapter "CA IDMS DMLO Sessions"

### CLIST Example #1

This CLIST deletes a specified record.

```
EXEC CALCDEL record-name calc-field calc-key del-opt
```

```
(1) CINIT %1
(2) MOVE %3 to %2
(3) OBTAIN CALC %1
(4) ON ANYERR QUIT REQUESTED RECORD NOT FOUND
(5) IFC %NULL4
    ERASE %1
    ELSE
    ERASE %1 %4 MEMBERS
    ENDIF
(6) ENDC
```

(1) Conditionally BIND and load the dictionary elements of record  
(2) Set *calc-field* to *calc-key*  
(3) Obtain the selected record (*record-name*)  
(4) If any CA IDMS error, stop execution and display 'REQUESTED RECORD NOT FOUND'.  
(5) Determine format of CA IDMS DML ERASE command based on content (or NULL status) of *del-opt*.  
(6) Required CLIST terminator

## CLIST Example #2

This CLIST sweeps an area or set, looking for a field value in a record. It makes use of variable processing/prompting and conditional statements.

**EXEC SCANFOR record set/area search-fl'd search val max-rec**

- (1) \*\*\*\* SWEEP AREA OR SET LOOKING FOR FIELD VALUE IN A RECORD
  
- (2) :%1 = PRIMARY RECORD NAME  
:%2 = SET OR AREA NAME TO BE SWEPT  
:%3 = FIELD NAME TO BE TESTED  
:%4 = VALUE TO LOCATE  
:%5 = MAX RECORDS TO CHECK W/O HIT  
\*
  
- (3) OBTAIN FIRST %1 IN %2
  
- (4) REPEAT %5
  
- (5) ON 0307 EXITR
  
- (6) IFC %3 = %4  
EXITR  
ENDIF
  
- (7) OBTAIN NEXT %1 IN %2
  
- (8) ENDR
  
- (9) ENDC

- (1) COMMENT line
- (2) Define lines (used during variable prompting)
- (3) Start search process
- (4) Start of loop
- (5) On end-of-set, leave loop
- (6) If item of interest found, leave loop
- (7) Get next record
- (8) Required loop terminator
- (9) Required CLIST terminator

**Note:** In order for SCANFOR to work, you must establish currency for the owner if a set is selected.

### CLIST Example #3

This CLIST is an example of an 'application' CLIST which incorporates nesting of CLISTs. For a specified R-CUSTOMER record, locate the first R-PO (in CUST-PO set) for which PO-DATE is a specified value. If for that record, PO-UNSHIP = 0.00, delete the record.

**EXEC DELZERO 'SMITH' 890501**

- (1) \*\*\*\* DELETE SELECTED R-PO RECORDS
  - (2) :%1 = R-CUSTOMER RECORD KEY (MAX 20 BYTES)  
:%2 = PO-DATE VALUE TO BE LOCATED (YYMMDD)  
\*
  - (3) CINIT R-CUSTOMER
  - (4) MOVE %1 TO RCUST-KEY
  - (5) OBTAIN CALC R-CUSTOMER
  - (6) ON DB-REC-NOT-FOUND QUIT RECORD NOT FOUND
  - (7) EXEC SCANFOR R-PO CUST-PO PO-DATE %2 UNTIL
  - (8) ON DB-END-OF-SET QUIT NO PO FOR %2
  - (9) IFC PO-UNSHIP = 0.00  
    ERASE R-PO ALL MEMBERS  
ELSE  
    QUIT PO HAS NONZERO UNSHIP  
ENDIF
  - (10) ENDC
- 
- (1-2) COMMENT / DEFINES
  - (3) Ensure R-CUSTOMER field definitions loaded, and bound
  - (4) Set desired record key
  - (5) Read record of interest
  - (6) If not found, exit with message
  - (7) Call CLIST SCANFOR to walk CUST-PO set (see CLIST Example #2)
  - (8) If not found, exit with message
  - (9) Delete record or exit with message depending on PO-UNSHIP
  - (10) Required CLIST terminator





# Chapter 3: Commands

---

This section contains the following topics:

[Syntax Processor](#) (see page 58)

[Syntax Notation](#) (see page 58)

[CA IDMS DML Commands and Restrictions](#) (see page 59)

[CA IDMS DML Control Statements](#) (see page 60)

[CA IDMS DML Retrieval Statements](#) (see page 61)

[CA IDMS DML Modification Statements](#) (see page 65)

[CA IDMS DML Save Statements](#) (see page 66)

[LRF Commands](#) (see page 67)

[CA IDMS DML SCRATCH/QUEUE Processing](#) (see page 69)

[CA IDMS DML Command Restrictions](#) (see page 73)

[CA IDMS DMLO Commands](#) (see page 74)

[PROFILE Maintenance Commands](#) (see page 91)

[CLIST Maintenance Commands](#) (see page 95)

[CLIST Commands](#) (see page 98)

[PF Key Processing](#) (see page 106)

[Variable Symbols](#) (see page 107)

[Extended Command Processing](#) (see page 111)

[Command Processing Examples](#) (see page 115)

## Syntax Processor

CA IDMS DMLO includes a powerful syntax processor which allows you to combine command-line entry, variable symbols, PF key text, and abbreviations (EQUATEs) to create executable commands. This chapter covers the following command-related topics:

- CA IDMS DML Commands
- LRF Commands
- SCRATCH/QUEUE Processing Commands
- CA IDMS DMLO Commands
- PROFILE Maintenance Commands
- CLIST Maintenance Commands
- CLIST Commands
- PF Key Processing
- Variable Symbols
- Extended Command Processing
- Command Processing Examples

See [CLIST Editing Commands](#) (see page 181) for a complete description of the CLIST editing commands.

## Syntax Notation

### Notation Conventions

Example	Function
OBTain CALC record-name	Keywords appear in mixed case. The minimum required portion of each keyword appears in uppercase.
STORE record-name	Variables appear in lowercase. You substitute an appropriate value for each variable.
DISPLAY record-name [CONTINUE]	Brackets indicate optional clauses or commands.
SET element < HEX > \ NATIVE /	Braces enclose two or more options. You select one of them.

Example	Function
Next ◀	A left arrow indicates the CA IDMS default.

## CA IDMS DML Commands and Restrictions

CA IDMS DMLO supports all CA IDMS DML commands except for those exceptions listed below.

This guide assumes that you are generally familiar with CA IDMS DML commands and understand the concept of CA IDMS currency. See the appropriate CA IDMS manuals for detailed information on these topics.

## CA IDMS DML Control Statements

The following is a list of CA IDMS DML control statements in the formats accepted by CA IDMS DMLO:

```

    r/          \r
    BIND |< record-name >|
        L\ ALL      /L
    
```

```

    READY [area-name] |
    | USAGE-mode [IS] |< PROTeCted >| < RETRIeval >|
    |                  L\ EXCLusive /L \ UPDate      /L
    
```

COMMIT [TASK] [ALL]

```

    r/ NOSAVE          \r
    FINISH [TASK] |< SAVE          >|
                L\ SAVE=profile-name /L
    
```

```

    r/ CONTinue        \r
    ROLLBACK [TASK] |< NOSAVE          >|
                  || SAVE          ||
                  L\ SAVE=profile-name /L
    
```

IF set-name IS [NOT] EMPTy [imperative-statement]

IF [NOT] set-name MEmber [imperative-statement]

KEEP [EXCLusive] CUrrent record-name

```

    /          \ /          \
    KEEP [EXCLusive] CUrrent < WIThin > < set-name >
    \ IN      / \ area-name /
    
```

## CA IDMS DML Retrieval Statements

The following is a list of CA IDMS DML retrieval statements in the formats accepted by CA IDMS DMLO:

### Format 1

```
< FIND > [KEEP [EXCLUSIVE]] [record-name] DB-KEY [IS] < KEYn >
\ OBTain /                               \ dbkey-literal /
```

### Format 1B

```
< FIND > [KEEP [EXCLUSIVE]] DB-KEY [IS] < KEYn >
\ OBTain /                               \ dbkey-literal /
  PAGE-INFO < PGRn >
              \ pgr-literal /
```

### Format 2

```
< FIND > [KEEP [EXCLUSIVE]] Current record-name
\ OBTain /

< FIND > [KEEP [EXCLUSIVE]] Current < WITHin > < set-name >
\ OBTain /                               \ IN / \ area-name /
```

### Format 3

```

/ Next \
| PRIor |
< FIND > [KEEP [EXCLUSIVE]] < First > [record-name]
\ OBTain /                               | Last |
                                          \ number /

/ WITHin \ / set-name \
< IN > < area-name >
```

### Format 4

```
< FIND > [KEEP [EXCLUSIVE]] Owner < WITHin > set-name
\ OBTain /                               \ IN /
```

### Format 5

```

/ CALC \
< FIND > [KEEP [EXCLUSIVE]] < ANY > record-name
\ OBTain /                               \ DUPLICATE /
```

### Format 6

```
< FIND > [KEEP [EXCLUSIVE]] record-name < WITHIN > area-name
\ OBTain /
                                     \ IN /
                                     / SORTKEY \
set-name [CURRENT] USING < field-name >
                                     \ LITERAL /
Get
GET record-name
```

## Using FIND/OBTAIN Commands

In the FORMAT 3 FIND/OBTAIN the clause 'WITHIN set-name/area-name' may be omitted. If omitted, this will default to 'WITHIN area-name' and CA IDMS DMLO will determine the correct area-name for the record requested.

FORMAT 1, 5, and 6 of the FIND/OBTAIN commands require definition of a symbolic key or database key before execution. The following discussion explains how to execute these commands in the CA IDMS DMLO environment.

### Format 1

```
< FIND > [KEEP [EXCLUSIVE]] [record-name] DB-KEY [IS] < KEYn >
\ OBTain /
                                     \ dbkey-literal /
```

The Format 1 FIND/OBTAIN command requires a database key. The db-key value may be provided as a literal or in one of the ten CA IDMS DMLO KEYPAD fields. The *dbkey-literal* may be specified as Pppppp-Llll (page-line format), X'hhhhhhh' (hexadecimal format), or F'nnnnnnnn' (decimal format).

A value may be established in the KEYn KEYPAD by using one of the following commands:

1. ACCEPT KEYn FROM ..... CURRENCY
2. RETURN KEYn FROM .....
3. MOVE ..... TO KEYn

## Format 1B

```
< FIND > [KEEP [EXCLUSIVE]] DB-KEY [IS] < KEYn > PAGE-INFO < PGRn
>
\ OBTain / \ dbkey-literal / \ pgr-literal
/
```

The PAGE-INFO value may be provided as a literal or in one of the ten CA IDMS DMLO PGR0 - PGR9 KEYPAD fields. The pgr-literal may be specified as X'hhhhhhh' (hexadecimal format) or F'nnnnnnn' (decimal format).

A value may be established in the PGRn KEYPAD by using one of the following commands:

```
ACCEPT PGRn FOR record-name
ACCEPT KEYn FROM ... CURRENCY
```

## Format 5

```
< FIND > [KEEP [EXCLUSIVE]] < ANY > record-name
\ OBTain / \ DUPLICATE /
```

To execute a FORMAT 5 FIND/OBTAIN CALC command you must set the record elements comprising the calc key. To accomplish this use one of the following procedures:

- **Using the Data Manipulation Screen**
  1. Display the record type to be retrieved.
  2. Enter appropriate data in the calc key field(s).
  3. Press the ENTER key.
  4. Enter the FORMAT 5 command in the command area.
  5. Press the ENTER key or a PF key.
- **Using the Command Line (Option 1)**
  1. MOVE key-value TO sort-key-field (for each element of sort key)
  2. Enter the FORMAT 6 command and press the ENTER key or a PF key.
- **Using the Command Line (Option 2)**
  3. Enter the format 6 command with an appropriate literal and press the ENTER key or a PF key.

## Format 6

```
< FIND > [KEEP [EXCLUSIVE]] record-name < WITHIN > area-name
\ OBTain /
                                     \ IN /
                                     / SORTKEY \
set-name [CURRENT] USING < field-name >
                                     \ Literal /
```

The Format 6 FIND/OBTAIN USING command needs the symbolic key of a sorted set key to execute.

The method for executing the Format 6 statement is similar to the method used to execute the Format 5 statement. However, the Format 6 statement pertains to indexed records and sorted sets.

To execute this statement CA IDMS DMLO must be given the value of the sorted or index key.

To supply CA IDMS DMLO with the information, use one of the following methods:

- **Using the Data Manipulation Screen**
  1. Display the record type to be retrieved.
  2. Enter appropriate data in the sort key field(s).
  3. Press the ENTER key.
  4. Enter the Format 6 command and press the ENTER key or a PF key.
- **Using the Command Line (Option 1)**
  1. MOVE ... sort-key-field
  2. Enter the Format 6 command press the ENTER key or a PF key.
- **Using the Command Line (Option 2)**
  3. Enter Format 6 command with literal of appropriate type.



## FIND/OBTAIN Special Considerations

### Symbolic Keys

In the Format 6 FIND/OBTAIN command, there are three options for specifying the symbolic key:

- **SORTKEY** causes CA IDMS DMLO to gather multiple sort-keys into an internal work area for the use of the actual CA IDMS DML call to CA IDMS. The SORTKEY parameter provides support for multiple non-contiguous sort-key fields, but may also be used for records with contiguous or single sort-key fields.

If SORTKEY is entered for a set sorted by db-key, then the KEY field, as shown in the command/status area, will be used as the key for the CA IDMS DML call to CA IDMS.

- *field-name* names a field which will be used as the key for the set or index named in the command. This option is used for contiguous or single sort-key fields. The specified *field-name* need not be an element within the specified record.
- *literal* may be specified for any sorted set. The literal may be in character, hexadecimal, or dbkey format as may be appropriate. Use of a literal behaves as though the data had in fact been keyed into the record fields comprising the SORTKEY (literal length need not match the sum of fields).
- See the discussion of CLIST Example #2 for allowable literal formats.

## CA IDMS DML Modification Statements

The following is a list of CA IDMS DML modification statements in the formats accepted by CA IDMS DMLO:

**STOre record-name**

```

ERASE record-name [ / PERManent \ ]
                  | < SElective > |
                  [ \ ALL MEmbers / ]

```

**MODify record-name**

**CONNect record-name TO set-name**

**DISConnect record-name FROM set-name**

## CA IDMS DML Save Statements

The following is a list of CA IDMS DML save statements in the formats accepted by CA IDMS DMLO:

```

/          \
ACcEpt KEYn FROM / record-name \ CURRENCY /          \
                  \ set-name   /          \ PGRn /
                  \ area-name  /

```

```

/ NEXT \
ACcEpt KEYn FROM set-name < PRIOR > CURRENCY /          \
                  \ OWNER /          \ PGRn /

```

ACcEpt PGRn FOR record-name

```

/ IDMS-STATISTICS \
ACcEpt < STATISTICS > [RESET] [DELTA]
      \ STATS          /

```

```

/ Currency          \
| First [Currency] |
| Last [Currency]  |
RETURN KEYn FROM index-set-name < Next [Currency]          > [KEY [INto] field-
name]
| Prior [Currency] |
| USING index-key-value-v |
\ USING SORTKEY      /

```

## ACCEPT/RETURN Special Considerations

*Keyn* always refers to one of the ten CA IDMS DMLO KEYPAD fields.

*PGRn* always refers to one of the ten CA IDMS DMLO PAGE-INFO values.

In the RETURN statement:

- *index-key-value-v* may be an element within a previously referenced record, or may be a literal value
- *field-name* is any element within a previously referenced record.

In the ACCEPT IDMS-STATISTICS statement:

- RESET and DELTA are CA IDMS DMLO extensions to standard CA IDMS DML syntax which permit display of incremental statistics.
- If neither RESET nor DELTA is used, statistics values displayed are cumulative since the start of your CA IDMS DMLO session.
- If DELTA is used, the statistics represent the incremental change since RESET was last used as an ACCEPT parameter, or if RESET was never used, since the beginning of your session.
- If RESET is used, a new "zero" point is established for later DELTA usage.

## LRF Commands

The Logical Record Facility of CA IDMS DMLO allows access to both LR and mixed subschemas. All LRF commands are available, as well as some WHERE clause capabilities.

CA IDMS DMLO supports all WHERE clause features except:

- Subscripted fields
- More than 4 levels of qualification

WHERE clause syntax is described below.

The WHERE clause can be used in conjunction with the following LRF commands:

```
OBTAIN FIRST <logical-record-name> WHERE . . .
OBTAIN NEXT  <logical-record-name> WHERE . . .
MODIFY       <logical-record-name> WHERE . . .
ERASE       <logical-record-name> WHERE . . .
STORE       <logical-record-name> WHERE . . .
```

See the appropriate CA ADS or CA IDMS DML manual for additional information on these commands.

## Entering LRF Commands

Because LRF commands have the potential to be much longer than CA IDMS DML commands, a new optional format for the static part of the screen is available to deal with LRF commands. The new format may be selected using the SET LRF ON/OFF command.

CA IDMS DMLO, CA IDMS DML, and LR commands may always be entered in the MENU/ASSIST Mode or standard screen formats; the larger command area in the 'LR' format is available if needed.

To enter commands longer than the available command line(s), use the EQUATE phrase capability of CA IDMS DMLO to create a shorthand form of the command. The expanded command length may exceed the input area length.

**Note:** The LRF screen is applicable not only for LRF commands, but may be used whenever an expanded command area is needed.

## WHERE Clause

```
WHERE [NOT] expression1  $\left\{ \begin{array}{l} / \quad \backslash \\ < \text{AND} > \\ \backslash \text{OR} / \end{array} \right.$  [NOT] expression2]. . .
```

Where:

**expression**-DBA keyword or comparison

**dbkeyword**-DBA-designated keyword to select path

**comparison**-item1 relational item2

**relational**-MATCHES, CONTAINS, EQ, =, GT, >, LT, <, NE, GE, LE

**item**-literal

idd-defined-field-name [OF group-ID] ...

logical-rec-field-name [OF group-ID] ... [OF LR]

arithmetic expression involving any of the above

## CA IDMS DML SCRATCH/QUEUE Processing

The following is a list of CA IDMS DML scratch/queue processing statements in the formats accepted by CA IDMS DMLO.

See the appropriate CA IDMS DML manual for detailed information on scratch/queue syntax.

CA IDMS DMLO not only provides support for standard CA IDMS DML syntax for scratch/queue processing but also includes some powerful extensions which simplify your use of these commands. The extensions include the following:

- Abbreviations for most keywords
- SHOW SCRATCH command and selection variable (?T) to display those scratch areas associated with your signon
- Scratch/queue entity variables (&T and &Q)
- Default scratch/queue buffer areas
- Logic to assign data area and length when not included in the command.

### Abbreviations

The standard abbreviations and default parameters are described below in the detailed syntax for the scratch/queue processing commands.

### Scratch Display

The SHOW SCRATCH command (or ?T selection variable) allows you to list the scratch areas associated with your CA IDMS/DC terminal. You may also invoke this list by using the ?T selection variable.

### Scratch/Queue Entity Variables

You may always reference the most recent scratch area ID or queue ID by using the entity variables &T and &Q. These variables may appear anywhere a scratch area ID or queue ID are required.

## Default Buffer Areas

You may specify any subschema or dictionary-owned record (or elements within such records) as buffer areas for scratch/queue commands. You may also use SBUFn and QBUFn (n=0-9) which will be automatically allocated by CA IDMS DMLO. SBUF and QBUF are the default values for installation-defined prefixes. Consult your system administrator for the prefix chosen at your installation.

## Default Data and Length Assignment

You may omit specification of data area and/or length for a GET or PUT command. CA IDMS DMLO attempt to establish a logical value for the omitted parameter. The rules for establishing such values are as follows:

- If you specify a record name as a data area, the default length will be the record length.
- If you specify a field name as a data area, the default length will be from the start of that field to the end of its record.
- If you omit the data area from a PUTSCR request, the default will be the area and length used in a prior PUT for the same scratch area ID.
- If you omit the data area from a GETSCR request, the default area and length will be selected from:
  - Prior GETSCR request for the same scratch area ID
  - Prior PUTSCR request for the same scratch area ID
  - Unused SBUFn default buffer name
- If you omit the data area from a PUTQUE request, the default will be the area and length used in a prior PUT for the same queue ID.
- If you omit the data area from a GETQUE request, the default area and length will be selected from:
  - Prior GETQUE request for the same queue ID
  - Prior PUTQUE request for the same queue ID
  - Unused QBUFn default buffer name

**GET SCRatch**

[[AREA] [ID] scratch-area]

```

┌ / Next◀           \  ┘
| | First           |  |
| < Last           >  |
| | Prior           |  |
| | Current         |  |
└ \ REcOrd ID scr-rec-id /  ┘

```

```

/           \
< Delete◀ >
\ Keep     /

```

```

┌                               ┌/                               \┐┘
| INto buff-start |< TO buff-end           >||
└                               \ \ MAX Length max-reclen /┘┘

```

[RETurn LENgth [INto] len-field]

**PUT SCRatch**

[[AREA] [ID] scratch-area]

```

┌                               ┌/                               \┐┘
| FROM buff-start |< TO buff-end           >||
└                               \ \ Length scr-recLen /┘┘

```

[REcOrd ID scr-rec-id [REPlace]]

[RETurn REcOrd ID [INto] rid-field]

**DELETE SCRatch**

[[AREA] [ID] scratch-area]

```

┌ / Next           \  ┘
| | First           |  |
| < Last           >  |
| | Prior           |  |
| | Current◀       |  |
└ \ REcOrd ID scr-rec-id /  ┘

```

[RETurn REcOrd ID [INto] rid-field]

**GET QUEUE**

[[ID] queue-area]

```

┌ / Next◀           \ ┐
│ | First           | │
│ < Last           > │
│ | Prior           | │
│ | N th           | │
└ \ REcOrd ID scr-rec-id / ┘

```

```

/           \
< Delete◀ >
\ Keep     /

```

```

/           \
< L0ck◀ >
\ NOLock  /

```

```

/           \
< Wait    >
\ NOWait◀ /

```

```

┌                               ┐ /           \ ┐
| INTO buff-start |< TO buff-end           >||
└                               ┘ \           / ┘
┌                               ┐ \           / ┘
└                               ┘ \ MAX Length max-reclen / ┘

```

[RETurn LENgth [INto] len-field]

**PUT QUEUE**

[[ID] queue-area]

```

/           \
< FIrst >
\ Last  /

```

```

┌                               ┐ /           \ ┐
| FROM buff-start |< TO buff-end           >||
└                               ┘ \           / ┘
┌                               ┐ \           / ┘
└                               ┘ \ Length scr-reclen / ┘

```

[RETurn REcOrd ID [INto] rid-field]

[RETENTION retn-days]

**DELETE QUEUE**

[[ID] queue-area]

```

/           \
< Current◀ >
\ All     /

```



## CA IDMS DML Command Restrictions

Almost all of the CA IDMS DML commands are supported by CA IDMS DMLO. However, there are a few exceptions. The CA IDMS DML commands that are not supported or supported with restrictions are described below:

- These commands are *not* supported by CA IDMS DMLO:

```
ACCEPT bind-address-v FROM record-name BIND
ACCEPT proc-control-location-v FROM proc-name PROCEDURE
BIND PROCEDURE FOR proc-name TO proc-control-location-v
```

- In the Format 3 FIND command, the integer must be explicit; no variable can be referenced. For example:

```
FIND 4 ORDER WITHIN CUSTOMER-ORDER
```

- The IF command can be entered with or without a second compound command. When the IF command is entered as a single command the appropriate error-status and the message TRUE or FALSE is displayed. When the expression is true, the second command (if present) will be executed.
- The ACCEPT and RETURN commands normally refer to the user-defined database key and page-info identifiers. In CA IDMS DMLO, database key identifiers are of the form KEY0 - KEY9. KEY is equivalent to KEY0. Page-info identifiers are of the form PGRO-PGR9. Contents of these fields can be displayed with the SHOW KEYPADS command.
- The FIND DB-KEY IS command may specify one of the KEY identifiers or a database key literal in one of three acceptable formats are:

```
PPPPP-LLLL (page-line format)
X'hhhhhhh' (hexadecimal format)
F'nnnnnnn' (decimal format)
```

### Examples

```
OBTAIN <record> DB-KEY 20510-11
OBTAIN <record> X'00501E0B'
OBTAIN <record> F'5250571'
OBTAIN <record> KEY3
```

A value may be established in a KEYn field by using ACCEPT, RETURN, MOVE, or by specifying those clauses of scratch and queue commands which return RECORD IDs.

- ACCEPT db-statistics-v FROM IDMS-STATISTICS has been changed to ACCEPT IDMS-STATISTICS.
- The BIND RECORD command syntax has been extended to include BIND and BIND ALL. The BIND command causes the Record Display screen to appear. If BIND ALL is entered, all of the records in the subschema are bound. If BIND RECORD is entered, only the specified record will be bound.

- CA IDMS DML READY command syntax is fully supported. Note that READY issued with no optional clauses readies all areas in a subschema in shared retrieval usage mode.

Areas can also be readied on the Area Display screen. Access the Area Display screen by using the CA IDMS DMLO READYLIST, SHOW AREAS, or ?A commands.

- CA IDMS DMLO performs complete syntactical checks on all commands entered. For example, CA IDMS DMLO will verify that area, record, and set names used in commands are valid (i.e., they exist in the subschema).

CA IDMS will check the semantics of all commands and the CA IDMS error status returned in the command area will indicate the nature of any semantic errors.

- A field with multiple subscripts is displayed with only its low-order subscript. Higher subscripts are displayed on the high-order occurring group.

## CA IDMS DMLO Commands

CA IDMS DMLO commands are entered in the command area. These commands are summarized below.

Command	Function
DISPLAY	Display a record, group, or element within a record using various display options.
EDIT	Create or edit a CLIST.
EQUATE	Establish abbreviations for any keyword, phrase, or subschema entity.
EXEC	Execute a CLIST.
HELP	Access CA IDMS DMLO online documentation.
INIT	Load record description from the dictionary and/or initialize buffer.
LISTC	Access the CLIST Selection/Maintenance screen.
LEARN	Activate Learn Mode during CLIST Step Mode execution.
MOVE	Transfer data.
PRINT	Print the complete current formatted display or the currently displayed screen.
READYLIST	Display the Area Selection screen.
SAVE	Display the PROFILE Save screen for intermediate PROFILE saves.

Command	Function
SET	Set various session values.
SHOW	Display various session parameters and subschema entities.

## DISPLAY Command

```
DISplay  r r / record-name \ r / CONTINUE \ r r
          | | < SUBSCHEMA-CTRL > | | | HEX | | | | | | |
          | | \ SSCTRL / | | | < NATive > | | |
          | | | | | | | | | DUMP | | |
          | | | | | | | | | \ NODUMP / | | |
```

```
DISplay field-name r / OF \ record-name r r / HEX \ r
                  | < WITHin > | | | NATive > | | |
                  | | IN | | | DUMP | |
                  | | \ NODUMP / | | \ | / |
```

The DISPLAY command displays a record, group, element within a record, or field. Each record has an I/O buffer that can be displayed independently of any CA IDMS DML command.

You can display a record or field in hex or native, or use the DUMP option to display a record or field in side-by-side hex and character formats. The DUMP option overrides, but does not replace, the current HEX/NATIVE setting for the specified record or field. The NODUMP option causes the display option to revert to HEX or NATIVE, whichever was active when DUMP was specified.

When using the DUMP display option, you can make updates in either format at the same time. Data changes are processed left-to-right as follows:

1. Any changed hex data is applied to the record/field buffer.
2. Any character data which differs from the original buffer contents is applied to the record/field buffer.
3. The changed line is completely reformatted.

A group or element can be displayed to initialize a key field prior to obtaining a calc, indexed, or sorted record. CA IDMS DMLO assigns a unique number to each FILLER field (e.g., FILLER#004). The optional 'WITHIN *record-name*' clause can be used to request a non-unique element or group name. The DISPLAY command can also display the current contents of the Subschema-Control block. DISPLAY *record-name* CONTINUE allows you to begin formatting at the point in the record where storage was exhausted.

## DISPLAY Command Rules

1. Group and element names cannot be subscripted.
2. A record display may exhaust available CA IDMS DMLO display storage before all fields (or occurrences) are formatted. A warning message is issued to inform you of this condition. DISPLAY CONTINUE allows you to begin formatting at the point in the record where storage was exhausted.
3. CONTINUE may only be requested for the most recently OBTAINed or DISPLAYed record.
4. If neither *field-name* nor *record-name* is specified, the most recently displayed record will be redisplayed according to and format parameter specified.

## EDIT Command

**EDit** [**S.**]clist-name

Where:

**S.**-forced access to a global CLIST when there is a global CLIST with the same name as a CLIST associated with your user ID.

**clist-name**-1-8 character CLIST identifier.

Use the EDIT command to create or edit a CLIST. This invokes an editor session. See [CLIST Editing Commands](#) (see page 181) for more information on CLIST editing commands.

You may also request an EDIT session for a CLIST by entering "E" next to an item on the CLIST Selection/Maintenance screen.

## EQUATE Command

```

      / old-word \
Equate new-word < phrase >
      \ null    /

```

Where:

**new-word**-alphanumeric token which begins with a letter, and does not include an "&", "@", or "%".

**old-word**-any word currently defined to the CA IDMS DMLO session. This includes commands, keywords, literals, area-names, record-names, set-names, field-names, or other EQUATE symbols.

**phrase**-an arbitrary set of tokens which may include any symbols eligible for *old-word* as above, as well as macro variable symbols ("&" and "@").

**NULL**-removes the association of *new-word* from the list of session EQUATEs. You can re-use *new-word* during the session.

Use the EQUATE command to establish convenient abbreviations for frequently used entities, commands, or partial commands. Some examples of EQUATE usage are listed below.

See the discussion of Extended Command Processing for a more detailed description of the iteration of PF keys, EQUATEs, variables, and command specification.

## EQUATE Command Examples

### EQUATE EMP EMPLOYEE

This example establishes a simple abbreviation for the record EMPLOYEE. Anywhere the record-name could be used, the EQUATE symbol EMP may appear.

### EQUATE OFR OBTAIN FIRST

This example establishes an EQUATE symbol for a partial command. You could enter OFR IN EMP-DEMO-REGION as an acceptable CA IDMS DML command.

### EQUATE ONWR OBTAIN NEXT &1 WITHIN &2

This example establishes an EQUATE symbol for partial command. You could enter ONRW EMP &A to result in an effective CA IDMS DML command of OBTAIN NEXT EMPLOYEE WITHIN EMP-DEMO-REGION.

## EQUATE Command Rules

1. EQUATE symbols you define during a session may be preserved in a PROFILE when you end the session. They will then be re-established at the start of subsequent session if you request that PROFILE.
2. EQUATEs may be nested three deep; i.e., you may EQUATE a symbol to a phrase which includes EQUATE symbols, and some of those symbols may be phrases which include EQUATE symbols which reference phrases.
3. EQUATE symbols may be incorporated in PF/PA key text.
4. CLIST variable symbols (%n) may not be contained within EQUATE phrases or as part of EQUATE symbols.

## EXEC Command

```
EXEC [S.]clist-name < A1. . .A9 >  
          \ ?          /
```

Where:

**S.**-forced access to a global CLIST when there is a global CLIST with the same name as a CLIST associated with your user ID.

**clist-name**-1-8 character CLIST identifier.

**a1...a9**-represent optional arguments used for macro-like replacement within the body of the CLIST. These arguments may be keywords, subschema entity names, or literals of any acceptable format. Within the CLIST, *a1* will replace each occurrence of %1, etc.

**?**-causes the CLIST processor to prompt for all arguments and display all DEFINE information.

Use the EXEC command to execute an existing CLIST.

You may incorporate CLIST variables within any CLIST statements. These variables are of the form %1 ... %9. This variable capability allows you to prepare general-purpose modules for frequently used functions.

Arguments supplied with the EXEC statement may be record, area, set, scratch/queue names, element names, keywords, or literals of any acceptable format. The arguments will replace all occurrences of the corresponding variables within the CLIST statements at the time the statement is executed.

You may indicate that an optional argument is null by entering "," for that argument. This will cause the corresponding symbol to be replaced by a null string in the CLIST command. You may test if argument *n* is null with a conditional test for %NULLn. This conditional name is "true" if the corresponding argument is supplied as a null value.

The CLIST facility allows you to use DEFINE symbols (:%n) for documenting the use of CLIST arguments ahead of the executable CLIST commands. Not only will this provide documentation for a user who is editing the CLIST, but it allows the CLIST processor to prompt for missing arguments. If a CLIST command is encountered which includes a CLIST variable for which a corresponding argument has not been provided, the processor will display a screen which allows you to supply the missing argument value. If a DEFINE exists for the missing argument, the associated documentation is also displayed.

If you have no idea of what arguments are needed for a given CLIST, "EXEC *clist-name* ?" will cause the CLIST processor to prompt for all arguments, and display all DEFINE information. Such documentation for any non-trivial CLIST is highly recommended. Note that a CLIST may EXEC another CLIST. Thus, you may define CLIST subroutines called by other CLISTS. CA IDMS DMLO will ensure that recursion doesn't occur; i.e., a CLIST doesn't appear at multiple levels in an active EXEC nest.

## HELP Command

### HELP

Use the HELP command to access CAIDMS DMLO online documentation.

CA IDMS DMLO online documentation includes:

- Complete descriptions of all CAIDMS DMLO screens
- Comprehensive information concerning various CAIDMS DMLO processing topics
- New features summaries
- Online message facility. You can access the online message text by entering option M at the first screen of any online documentation module.

The CA IDMS DMLO online documentation print utility gives you the ability to create a structured hardcopy printout of any CA IDMS DMLO online documentation module, including CAIDMS DMLO messages. See [Commands](#) (see page 57) for instructions on using the online documentation print utility.

## INIT Command

```
INIT rec-name [ / nnnn \ ]  
              [ [Version] < Highest > | [REPL]  
              [ \ Lowest / ]
```

Where:

**nnnn**-record version number

**REPL**-force reload of the record element descriptions for dictionary-owned records

Use the INIT command to ensure that all subschema or dictionary-owned record element descriptions are available and to initialize all elements to null values appropriate to their usage: numeric fields to zero, and others to spaces.

Normally, CA IDMS DMLO will automatically issue an INIT for a record the first time that record appears in an CA IDMS DML command. However, if you need to refer to an element within that record before the record itself has been referenced, you must issue the INIT command to force the load of record elements from the dictionary.

Once a subschema record has been loaded, you cannot reload the elements from the dictionary. However, you can reload a dictionary-owned record to allow access to a different version, or to refresh the currently loaded version if an update to the dictionary has occurred since you started the CA IDMS DMLO session.

Dictionary-owned records may be used as CLIST work areas or as input/output areas for scratch and queue processing.

## LEARN Command

```
LEARN / ON \  
      < OFF >  
      \ /
```

Use the LEARN command to activate the Learn Mode of command input during CLIST Step Mode execution. This mode allows you to key in commands. The commands are then executed and inserted into the active CLIST at the current point. IF and REPEAT statements are not eligible for Learn Mode.

The Learn Mode gives you the ability to dynamically update CLISTs during your normal CA IDMS DMLO processing.

You can initiate Learn Mode before the CLIST is executed or at any time during CLIST execution. It can also be used within a CLIST itself.

CA IDMS DMLO displays (*LEARN*) on the screen to let you know that you are in Learn/Step Mode.



## LIST Command

### LIST

Use the LIST command to:

- During session startup to redisplay the PROFILE Maintenance screen.
- At any other time during the CA IDMS DMLO session mainline to access the CLIST Selection/Maintenance screen.

## MOVE Command

### MOVE source TO target

Where:

**source** and **target** are any of the following:

- Fields in records
- Subschema control entities
  - ERROR-STATUS
- CA IDMS DMLO keypads (KEY0-KEY9)
  - DBKEY
  - DIRECT-DBKEY
- Figurative constants
  - SPACES
  - ZEROS
  - LOW-VALUES
  - HIGH-VALUES
- Literals of all types
  - alpha
  - numeric
  - hex
  - database key
  - full word

Use the MOVE command to move data without having to display the records containing the target fields.

## PRINT Command

**PRINT [ALL] [optional-heading]**

Where:

**ALL**-Print the complete current formatted display

**optional-heading**-An optional heading line

Use the PRINT command to instruct CA IDMS DMLO to print the currently displayed screen or the complete current formatted display (ALL keyword specified).

### Print Command Rules

1. PRINT is valid in a TSO environment.
2. A valid PRINT command causes one of the following messages:  
**PRINT SUCCESSFUL (self explanatory)**  
**PRINT ERROR (open or allocation error)**  
If no message occurs, you are in an unsupported environment for the PRINT command (CICS).
3. CA IDMS DMLO does not verify that the current formatted display matches the actual contents of the database. CA IDMS DMLO prints field modifications even if a record has not been stored or modified on the database.
4. Maximum length for the optional heading is restricted to the remainder of the command area following the PRINT command.
5. PRINT can be issued only from screens which have a command line area.

## READYLIST Command

< READYList >

\ RL /

Use the READYLIST command to display the Area Display screen.

You can use the Area Display screen to check the ready status of areas in the subschema and to ready areas. The Area Display screen displays the following information:

- The names of the areas in the subschema
- The usage mode of each area that has been readied

This command is included only for compatibility with earlier versions of CA IDMS DMLO. It is equivalent to the SHOW AREAS command.

## SAVE Command

### SAVE

Use the SAVE command to display the PROFILE Save screen to perform intermediate PROFILE saves.

## SET AUTOBIND Command

```
[SET] < AUTOBind > |< ON >|
          \ ABIND / | \ OFF /|
```

Use the SET AUTOBIND command to specify requirements for BINDing subschema records during an CA IDMS DMLO session. AUTOBIND ON will cause CAIDMS DMLO to automatically issue a BIND record-name command the first time that record-name is referenced in a command. AUTOBIND OFF forces you to enter an explicit BIND command before you can use a record. Note that it is more efficient to use the AUTOBIND ON option rather than issuing BIND ALL or individual BIND commands.

If ON or OFF is not specified, the option is toggled. SET AUTOBIND corresponds to "AUTO BIND RECORDS" on the Options Screen.

```
[SET] < AUTOHex > |< ON >|
          \ AHEX / | \ OFF /|
```

Use the SET AUTOHEX command to automatically change the display format of any element whose data contents does not match its PICTURE and/or USAGE to hexadecimal. Items with valid data are not affected.

If ON or OFF is not specified, the option is toggled.

If AUTOHEX is specified while a record is being displayed, enter the DISPLAY command to change the fields on the current display. SET AUTOHEX corresponds to "AUTOHEX DATA DISPLAY" on the Options Screen.

## SET CLIST Command

```
[SET] < CLIST > |< FAST >|
                | \ STEP /|
```

Use the SET CLIST command to change the execution mode for execution of CA IDMS DMLO CLISTs. If FAST or STEP is not specified, the option is toggled.

SET CLIST corresponds to "FAST CLIST EXEC MODE" on the Options Screen.

FAST mode execution will execute CLIST commands continuously until a PAUSE, PROMPT, QUIT, or ENDC command is encountered. STEP mode execution executes one CLIST command at a time. You must press ENTER after each command to continue execution. The next command in the CLIST will be displayed in the message display area.

**Note:** Display of status and error messages takes precedence over pending command display.

## SET CMDDISPLAY Command

```
[SET] CMDDisplay |< I >|
                 | \ U /|
```

Use the CMDDISPLAY command to specify whether commands will be displayed as I(input) or U(used) -- i.e., after the expansion of variables.

If I or U is not specified, the option is toggled. SET CMDDISPLAY corresponds to "AS USED CMND DISPLAY" on the Options Screen.

### Example

As an example of this option, consider the following:

- PF3 has been set to OBT N &1 in &2
- TEACHER &A was keyed into the command line and PF3 pressed

If option "I" is in effect, the command line will be unchanged.

If option "U" is in effect, the command line will contain:

```
OBT N TEACHER IN &A
```

## SET COBDISPLAY Command

```

      /          \ r/      \r
[SET] < COBDisplay > |< ON >|
      \ COBOL    / \ OFF /

```

Use the SET COBDISPLAY command to change the display format to one with indentation and level numbers similar to the standard COBOL record layout.

CA IDMS DMLO displays the native and the display usage of the item.

If OFF is specified, the display format will be vertical. If ON or OFF is not specified, the option is toggled.

The SET COBDISPLAY corresponds to "COBOL DISPLAY FORMAT" on the Options Screen.

## SET DEFENTK Command

```
SET DEFEntk < ON or OFF >
```

Use the SET DEFENTK command to change the default ENTER key use, when the ENTER key alone is hit, with no other data being typed/overtyped on the command line. By default, hitting the ENTER key alone will clear the line. If DEFENTK is set to OFF, then the last command on the DMLO command line (if any) will be re-executed. This feature is useful, for example, when navigating a database to repeat FIND/OBTAIN NEXT/PRIOR records without having to overtype data on the command line.

This value is initially set at install time and may be subsequently altered by changing the source of parameter module USDTPARM and re-assembling/re-linking same. It can also be switched dynamically at run time with the above SET DEFENTK command.

## SET EXIT Command

```

      r/      \r      r/      \r
[SET] EXIT |< ON >| |< ALL >|
      \ OFF /r \ Verb /r

```

Use the SET EXIT command to turn on/off calls to the installed exit program, either completely or for specified CA IDMS DML verbs.

The ability to use the SET USEREXIT command is specified at product installation using the customization macro. See your system administrator if you have any questions.

If ON or OFF is not specified, the option is toggled.

## SET HEX/NATIVE Command

```
SET record-name / HEX \
    < NATive >
    | DUMP |
    \ NODUMP /

SET field-name r/ OF \ record-name r / HEX \
    | < WIthin > | < NATive >
    ^\ IN / ^ | DUMP |
    \ NODUMP /
```

Use the SET HEX/NATIVE command to explicitly change the display format of the specified entity to/from hexadecimal format or the subschema-defined format, or in side-by-side hex and character formats (DUMP option).

The command does not cause the record, group, or element to be displayed. The new mode takes effect only when you enter the DISPLAY command or when a new occurrence of the record is obtained from the database.

When you specify NATIVE at the element level, CA IDMS DMLO displays the level number and usages (as shown below) of the specified element in the message area.

The DUMP option overrides, but does not replace, the current SET HEX/NATIVE setting for the specified record.

The NODUMP option causes the display option to revert to HEX or NATIVE, whichever was active when DUMP was specified.

You can make updates in either format at the same time. Data changes are processed left-to-right as follows:

1. Any changed hex data is applied to the record buffer.
2. Any character data which differs from the original buffer contents is applied to the record buffer.
3. The changed line is completely reformatted.

## SET INVCHAR Command

[SET] INVCHAR C

Where "c" represents a character which will be substituted for any non-displayable character when DUMP is selected as the display option.

SET INVCHAR corresponds to "INVALID CHAR REPLACE" on the Options Screen.

CA IDMS DMLO Descriptions	Schema Description Usage	Schema Description Picture
GROUP	N/A	X(n)
FILLER	N/A	X(n)
BIT	BIT	X(n)
DISPLAY	DISPLAY	X(n)
SDISPLAY	DISPLAY	S9(n)
UDISPLAY	DISPLAY	9(n)
SPACKED	COMP-3	S9(n)
UPACKED	COMP-3	9(n)
SBINARY	COMP	S9(n)
UBINARY	COMP	9(n)

## SET LOWERCASE Command

```
[SET] LOWERcase  r/      \r
                  |< ON >|
                  L\ OFF /L
```

Use the SET LOWERCASE command to allow CA IDMS DMLO to accept lowercase data.

If ON or OFF is not specified, the option is toggled.

**Note:** In an CA IDMS/DC environment, this option will be ineffective unless the DC LTE will accept lowercase input; i.e., specified via the DCUF SET UPLOW command (or CA task code CAPS OFF).

## SET LRF Command

```
[SET] LRF  r/      \n
           |< ON  >|
           L\ OFF /J
```

Use the SET LRF command to change the display format to create a larger command area to allow for longer LRF command input.

If ON or OFF is not specified, the option is toggled. The SET LRF corresponds to "LRF EXPAND CMND AREA" on the Options Screen.

**Note:** You do not need to be using an LRF subschema for this command to be possible.

## SET MAPIN Command

```
[SET] MAPIN r/      \n
            |< FAST >|
            L\ STEP /J
```

Where:

**FAST**-Allow you to enter a command and change data in a record buffer in the same converse.

**STEP**-Do not allow command entry and data change in a record buffer in the same converse.

If neither FAST nor STEP is specified, the option is toggled. The SET MAPIN corresponds to "FAST DATA INPUT MODE" on the Options Screen.



## SET MENU Command

```
[SET] MENU  ⌈/      \⌋
            |< ON  >|
            ⌔\ OFF /⌔
```

Use the SET MENU command to select the Menu Mode operation of CA IDMS DMLO.

The Menu Mode provides the less experienced CA IDMS user with a friendlier, structured environment, as well as a tool by which user CA IDMS DML syntax can be learned.

The following features are provided with the CA IDMS DMLO Menu Mode:

- Structured presentation of CA IDMS DML commands
- User-tailorable menu
- Entity selection lists
- Instructional command echo.

If ON or OFF is not specified, the option is toggled. The SET MENU corresponds to "DML/O 'MENU' MODE" on the Options Screen.

## SHOW AREANAMES Command

```
[SHOW] ⌈      ⌋
        |< AREANAMES >|
        ⌔\ AREAS  /⌔
```

Use the SHOW AREANAMES command to display all areas in the subschema. CA IDMS DMLO responds by displaying the Area Display screen. An example of this screen is shown in [General Information](#) (see page 11). Entry of the selection variable ?A as the command results in the same effect as if a SHOW AREA NAMES command were entered.

## SHOW EQUATES Command

```
[SHOW] EQUATES
```

Use the SHOW EQUATES command to display all user-defined EQUATE symbols. CA IDMS DMLO responds by displaying the Equates Display screen. An example of this screen is shown in [Data Manipulation and Display](#) (see page 15).

## SHOW KEYPADS Command

**[SHOW] KEYPads**

Use the SHOW KEYPADS command to display all database key and page-information values. CA IDMS DMLO responds by displaying the Database Key Display screen. An example of this screen is shown in [PF Key Usage](#) (see page 16).

## SHOW OPTIONS Command

**[SHOW] OPTions**

Use the SHOW OPTIONS command to display the Options screen. The Options screen is used to set various session parameters specified on the Signon screen or using various options of the SET command. An example of this screen is shown in [CLIST Programming](#) (see page 15).

## SHOW PFKEYS Command

**[SHOW] PFKeys**

Use the SHOW PFKeys command to display the current PF key settings on the PF Key Display screen. An example of this screen is shown in [Comprehensive Online Documentation](#) (see page 15).

For more information about using the PF Key Display screen, see the [Comprehensive Online Documentation](#) (see page 15).

## SHOW RECORDNAMES Command

```
[SHOW]  |< RECoRDnames >|
        | \ RECS      /|
```

Use the SHOW RECORDNAMES command to display all records in the subschema. CA IDMS DMLO responds by displaying the Record Display screen.

Entry of the selection variable ?R as the command results in the same effect as if a SHOW RECORDNAMES command were entered.

## SHOW SCRATCH Command

**[SHOW] SCRatch**

Use the SHOW SCRATCH command to display all scratch areas. CA IDMS DMLO responds by displaying the Scratch Display screen. An example of this screen is shown in [Selection Lists](#) (see page 16).

Entry of the selection variable ?T as the command results in the same effect as if a SHOW SCRATCH command were entered.

## SHOW SETNAMES Command

```
[SHOW]  |< SETNAMES >|
        | \ SETS      /|
```

Use the SHOW SETNAMES command to display all sets in the subschema. CA IDMS DMLO responds by displaying the Sets Display screen. An example of this screen is shown in EQUATE Facility.

Entry of the selection variable ?S as the command results in the same effect as if a SHOW SETNAMES command were entered.

## SHOW VARIABLES Command

**[SHOW] VARIables**

Use the SHOW VARIABLES command to display the current values of subschema variables. CA IDMS DMLO responds by displaying current value of the variables in the message area.

## PROFILE Maintenance Commands

PROFILE maintenance commands can only be entered on the PROFILE Selection/Menu screen. PROFILE maintenance command syntax can be divided into the following groups:

- **User Maintenance Commands**-allow maintenance of PROFILES associated with the signon user ID.
- **System Administrator Maintenance Commands**-allow the system administrator to act upon PROFILES of any/all users of the system.

## User PROFILE Maintenance Commands

User PROFILE maintenance commands include:

- LIST
- KILL
- COPY
- REPL
- RENAME

### LIST Command

**LIST [userid]**

Use the LIST command to refresh the list of available PROFILES. By specifying a *userid*, you can view the PROFILES associated with another user. The list includes global PROFILES.

### KILL Command

**KILL profile-name**

Use the KILL command to delete a PROFILE associated with your user ID. You may also delete a PROFILE by entering a "K" next to a PROFILE associated with your user ID on the PROFILE Maintenance screen.

After entering the KILL command, enter the LIST command to refresh the display.

## COPY/REPL Command

```
[ < COPY > [S.]profile1 [OF user1] [TO profile2]
[ \ REPL / ]
```

Where:

**S.**-forces access to globally-owned *profile1* if you have a PROFILE of the same name.

**profile1**-name of source PROFILE.

**OF USER1**-specifies a user ID other than your own.

**TO profile2**-specified target PROFILE name. The default is source name.

**REPL**-must be used to overlay a pre-existing PROFILE whose name matches the target name in the request.

Use the COPY command to copy a PROFILE.

After entering this command, enter the LIST command to refresh the display.

## RENAME Command

```
RENAME profile1 TO profile2
```

Use the RENAME command to change the name of a PROFILE associated with your user ID.

After entering this command, enter the LIST command to refresh the display.

## System Administrator PROFILE Maintenance Commands

System administrator PROFILE maintenance commands include:

- KILL ALL
- RENAME ALL

## KILL ALL Command

```
KILL ALL PROFILE < OF userid          >  
                \ BY DATE mm/dd/yy /
```

Where:

**OF userid**-specifies a mass delete of all PROFILES associated with the specified *userid*.

**BY DATE mm/dd/yy**-specifies the mass delete of all PROFILES which have not been accessed since the specified date.

Use the KILL ALL command to delete all the PROFILES associated with a user ID or all the PROFILES that have not been accessed since the specified date.

**Note:** CA IDMS DMLO updates a PROFILE with the current date whenever it is selected for a session. The BY DATE option allow you to purge dormant PROFILES. You may enter "U" next to a PROFILE on the selection list to force the last-used date to be updated to the current date. This allows you to quickly protect a set of PROFILES from BY DATE deletion.

## RENAME ALL Command

```
RENAME ALL user1 TO user2 [REPL]
```

Use the RENAME ALL command to remove a PROFILE from ownership of *user1* and associate it with *user2*. If *user2* has a PROFILE of the same name as *user1*, that PROFILE remains with *user1* unless the REPL (replace) option is specified. If *user2* doesn't exist, it is created.

**Note:** The system administrator also has the ability to execute the "LIST *userid*" command, and delete PROFILES directly from the list by entering "K" next to the selected items. Users other than the system administrator may issue the LIST *userid* command, but may not perform any update/delete functions against them.

## CLIST Maintenance Commands

CLIST maintenance commands may be entered at any time after the session startup procedure is complete. Normally they are entered in the command line of the CLIST Maintenance screen. Use the LIST command to display the CLIST Maintenance screen.

CLIST maintenance command syntax can be divided into the following groups:

- **User Maintenance Commands** - allow maintenance of CLISTs associated with the signon user ID.
- **System Administrator Maintenance Commands** - allow the system administrator to act upon CLISTs of any/all users of the system.

### User CLIST Maintenance Commands

User CLIST maintenance commands include:

- LIST
- KILL
- COPY
- REPL
- RENAME

#### LIST Command

**LIST [userid]**

Use the LIST command to refresh the list of available CLISTs. By specifying a *userid*, you can view the CLISTs associated with another user. If no user ID is specified, the list includes global CLISTs.

#### KILL Command

**KILL cList-name**

Use the KILL command to delete a CLIST associated with your user ID. You may also delete a CLIST by entering a "K" next to a CLIST associated with your user ID on the CLIST Maintenance screen.

After entering the KILL command, enter the LIST command to refresh the display.

## COPY/REPL Command

```
┌  
|< COPY >| [S.]clist1 [OF user1]  
[TO clist2]  
└\ REPL /┐
```

Where:

**S.**-forces access to globally-owned *clist1* if you have a CLIST of the same name.

**clist1**-name of source CLIST/

**OF user1**-specifies a user ID other than your own.

**TO clist2**-specified target CLIST name. The default is source name.

**REPL** - must be used to overlay a pre-existing PROFILE whose name matches the target name in the request.

Use the **COPY** command to copy a CLIST.

After entering this command, enter the **LIST** command to refresh the display.

## RENAME Command

```
RENAME clist1 TO clist2
```

Use the **RENAME** command to change the name of a CLIST associated with your user ID.

After entering this command, enter the **LIST** command to refresh the display.

## System Administrator CLIST Maintenance Commands

System administrator CLIST maintenance commands include the following:

- KILL ALL
- RENAME ALL



## KILL ALL Command

```
KILL ALL CLIST < OF userid          >
                          \ BY DATE mm/dd/yy /
```

Where:

**OF userid**-specifies a mass delete of all CLISTs associated with the specified *userid*.

**BY DATE mm/dd/yy**-specifies the mass delete of all CLISTs which have not been accessed since the specified date.

Use the KILL ALL command to delete all the CLISTs associated with a *userid* or all the CLISTs that have not been accessed since the specified date.

**Note:** CA IDMS DMLO updates a CLIST with the current date whenever it is selected for a session. The BY DATE option allows you to purge dormant CLISTs. You may enter "U" next to a CLIST on the selection list to force the last-used date to be updated to the current date. This allows you to quickly protect a set of CLISTs from BY DATE deletion.

## RENAME ALL Command

```
RENAME ALL user1 TO user2 [REPL]
```

Use the RENAME ALL command to remove a CLIST from ownership of *user1* and associate it with *user2*. If *user2* has a CLIST of the same name as *user1*, that CLIST remains with *user1* unless the REPL (replace) option is specified. If *user2* doesn't exist, it is created.

**Note:** The system administrator also has the ability to execute the "LIST *userid*" command, and delete CLISTs directly from the list by entering "K" next to the selected items. Users other than the system administrator may issue the "LIST *userid*" command, but may not perform any update/delete functions against them.

## CLIST Commands

CLIST command syntax can be divided into the following groups:

- Control Commands
- Documentation Commands
- Iteration Commands
- Status Test Command
- Conditional Commands
- Data Manipulation Command
- Interrupt Commands
- Record Processing Commands

Additional instructions on using CLIST commands, including CLIST examples, can be found in [Enter Key Usage](#) (see page 20).

### CLIST Control Commands

CLIST control commands include:

- EXEC
- ENDC
- EXITC
- QUIT

#### EXEC Command

##### **EXEC**

Use the EXEC command to invoke execution of a CLIST. CLIST executions may be nested 9 deep. See [CLIST Execution](#) (see page 49) for more information on executing CLISTs.

#### ENDC Command

##### **ENDC**

Use the ENDC command to indicate the physical end of a CLIST module. If encountered during execution, it functions as a GOBACK; i.e., it returns to the next higher CLIST, or to the native CA IDMS DMLO session controller.

## EXITC Command

### EXITC

Use the EXITC command to provide for immediate exit from the currently active CLIST. EXITC behaves like a GOBACK.

## QUIT Command

### QUIT

Use the QUIT command to provide a POPTOP function. It has the effect of consecutive EXITC's for each level of active CLIST. Any text following the QUIT verb on the command line will be displayed in the message area. Thus, it provides a convenient "panic-abort" function.

## CLIST Documentation Commands

CLIST documentation commands are identified by symbols rather than by keywords. The documentation commands are:

- **COMMENT** identified by "\*" in column 1 of the CLIST command
- **DEFINE** identified by ":%n" in column 1 of the CLIST command

## COMMENT Command

A COMMENT is a non-executable CLIST command which you may use for any documentational purpose within the body of a CLIST. The only COMMENT of any significance to the CLIST processor is the first command of a CLIST. If this command is a COMMENT, it will appear on the CLIST Maintenance screen as part of the informational display.

## DEFINE Command

The DEFINE command is used to document the use of variable symbols within a CLIST. You may use one DEFINE command (:%n) for each CLIST variable (%n) appearing in the CLIST. The DEFINE command is not executable, but does have special significance to the CLIST processor. If you fail to supply a value (or NULL indicator) for a CLIST variable when you EXEC the CLIST, the CLIST processor will recognize that fact when it encounters a command using that CLIST variable (%n). The CLIST processor will construct the CLIST Command Prompt screen using the DEFINE command(s) for the missing variable(s) in the current command. You may then supply a value for the CLIST variable(s) which will be used until CLIST execution terminates.

## CLIST Iteration Commands

CLIST iteration commands include:

- REPEAT
- ENDR
- EXITR

### REPEAT Command

**REPEAT UNTIL**  
**REPEAT nnnn**

Where:

**UNTIL**-signifies the loop is indefinite (i.e. commands within the loop will indicate termination).

**nnnn**-an integral literal representing a specific number of iterations. REPEAT groups may be nested to a level of 9. Each group must begin with REPEAT and end with ENDR.

Use the REPEAT command to provide an iteration facility for the CA IDMS DMLO CLIST.

### ENDR Command

**ENDR**

Use the ENDR command to indicate the end of a REPEAT group.

### EXITR Command

**EXITR**

Use the EXITR command to cause the CLIST execution to continue with the first command following the next ENDR command; i.e., to immediately exit the current level of REPEAT.

## CLIST Status Test Command

### ON Command

**ON x....x s.....s**

Where:

**x....x**-one of the following:

- nnnn-4-digit CA IDMS ERROR-STATUS code
- mnemonic-equivalent of 4-digit code
- lrf-status-text to match LR-STATUS

**s.....s**-any CA IDMS DML or CA IDMS DMLO command (except IFC and REPEAT)

Use the ON command to provide an easy way to test the CA IDMS ERROR-STATUS or LR-STATUS.

### Examples

**ON 0307 EXITR**

(If end-of-set exit current loop.)

**ON ANYERR OBTAIN OWNER WITHIN EMP-POSITION**

(If error-status non-zero execute CA IDMS DML obtain.)

**ON DB-REC-NOT-FOUND QUIT SELECTED RECORD NOT FOUND**

(If record-not-found stop the CLIST processing and display message.)

**ON LR-NOT-FOUND QUIT SELECTED LR RECORD NOT FOUND**

(If specified LR not found, stop CLIST processing and display message.)

**ON LR-PROBLEM QUIT LR MISHAP TOOK PLACE**

(If DBA-defined path status occurs, stop CLIST and display message.)

### Standard Mnemonics Provided

The following are standard mnemonics. Additional mnemonics can also be established using the EQUATE command.

```
0000 DB-STATUS-OK
nnnn ANYERROR
0307 DB-END-OF-SET
0326 DB-REC-NOT-FOUND
1205 DB-DUP-ON-STORE
0805 DB-DUP-ON-MOD
0705 DB-DUP-ON-CONN
```

## CLIST Conditional Commands

CLIST conditional commands include:

- IFC
- ELSE
- ENDIF

These commands provide a conditional facility for the CA IDMS DMLO CLIST. Allowable formats are:

```
IFC conditional  
  true-stmt1  
  ...  
  true-stmtn  
ELSE  
  false-stmt1  
  ...  
  false-stmtn  
ENDIF
```

```
IFC conditional  
  true-stmt1  
  ...  
  true-stmtn  
ENDIF
```

Where:

**true-stmtx**-represents any CA IDMS DML or CA IDMS DMLO commands, including EXEC, IFC, and REPEAT.

**false-stmtx**-represents any CA IDMS DML or CA IDMS DMLO commands, including EXEC, IFC, and REPEAT.

**valid conditionals**-operand1, relational, operand2, or %NULLn

**valid relationals**- = < > EQ GT LT GE LE NE NG NL MATCHES CONTAINS

**valid operands**-all the following are valid operands:

- field-name [OF record-name]
- ERROR-STATUS (subschemata control)
- DIRECT-DBKEY (subschemata control)
- DBKEY (subschemata control)
- CA IDMS DMLO KEYPADS (KEY0 - KEY9)
- Character literal 'ABCDEFGH'

- Hexadecimal literal X'hhhhhhh'
- Numeric literal 123 or 12.345
- Fullword literal F'1234567'
- Database key literal pppppp-III
- Figurative constant HIGH-VALUES LOW-VALUES SPACES ZEROS
- %NULLn (only to test %n for null value)

## CLIST Data Manipulation Command

### MOVE Command

**MOVE source TO target**

Where:

source and target may be any of the following:

- Fields in records
  - ERROR-STATUS
- CA IDMS DMLO KEYPADS (KEY0-KEY9)
- Figurative constants
  - SPACES, ZEROS
  - LOW-VALUES, HIGH-VALUES
- Subschema control entities
  - DBKEY, DIRECT-DBKEY
- Literals of all types
  - alpha 'HERE IS A LITERAL' or 'J' 'JONES'
  - numeric -123.45
  - hexadecimal X'005A2E0B
  - database key 20510-11
  - fullword F'5250571'

The generalized MOVE command, although not a CLIST-only command, provides a powerful capability for CLIST execution, and thus is included here.

## CLIST Interrupt Commands

CLIST interrupt commands include:

- PAUSE
- PROMPT
- RESUME

### PAUSE Command

**PAUSE *display-string***

Where:

***display-string***-any displayable string

Use the PAUSE command to provide for a temporary interruption of the CLIST execution.

Execution will temporarily be suspended and the *display-string* will be displayed in the message area. Execution will restart whenever the ENTER key is pressed with no command entry. You may enter commands while CLIST execution is suspended.

### PROMPT Command

**PROMPT *display-string***

Where:

***display-string***-any displayable string

Use the PROMPT command to provide for a temporary interruption of the CLIST execution.

Execution will temporarily be suspended and the *display-string* will be displayed in the message area. Execution will restart only when RESUME is entered on the command line. Until such time, commands and data may be freely entered.

### RESUME Command

**RESUME**

Use the RESUME command to restart CLIST execution after a PROMPT command is encountered.

**Note:** RESUME may not be part of a CLIST; it is only for manual entry on the command line.



## CLIST Record Processing Commands

CLIST record processing commands include:

- CBIND
- CINIT

### CBIND Command

**CBIND** **record-name**

Where:

**record-name**-any record name

Use the CBIND command to provide for a conditional BIND of a subschema record if it has not previously been bound.

### CINIT Command

**CINIT** **record-name**

Where:

**record-name**-any record name

Use the CINIT command to provide for a conditional loading of record elements from the dictionary.

CINIT initializes the buffer of the named record. If necessary, the record will be bound, and the elements of the record will be loaded from the dictionary.

## PF Key Processing

CA IDMS DMLO allows commands or partial commands to be assigned to PF keys. This can greatly reduce the command data entry requirements during an CA IDMS DMLO session. PF keys assignments may be established in the following ways:

- Your data administrator may assign default PF key values during product installation.
- You can press a PF key (rather than the ENTER key) immediately after keying a command. This will assign the keyed command text to the PF key.
- You can enter the SHOW PFKEYS command to display all PF key settings, and enter text directly on the screen.

PF key settings can be preserved as part of the session PROFILE.

The only formal restriction as to what text may be associated with a PF key is the restriction against using CLIST variable symbols (%1-%9) within the text. PF keys may contain EQUATE symbols and full or partial commands, and may incorporate variable symbols.

Length of the text associated with a PF key is initially set to the physical screen width. If a longer text area is required (probably in an LRF context), it may be established by keying in a long command on the command line and pressing the desired PF key. Available length will be expanded to the next higher multiple of screen width sufficient to contain the long text string. Once the text area has been expanded, you may use the PF Key Display screen to update whatever length has been allocated.

Text associated with a PF key is normally treated as though it had been keyed ahead of any command line text (unless the command line begins with an CA IDMS DML or CA IDMS DMLO verb, in which case the PF key is assigned to the command line text).

**Note:** For a detailed explanation of how the CA IDMS DMLO command processor uses PF key text, see [Extended Command Processing](#) (see page 111) and [Command Processing Examples](#) (see page 115).

## Variable Symbols

To provide flexible processing, CAIDMS DMLO recognizes the following types of variable symbols for use in constructing executable commands:

- Subschema variables
- Macro variables
- Positional variables
- Selection variables
- CLIST variables

## Subschema Variables

The subschema variables (&R, &D, &A, &S, &T, and &Q) can appear in any CA IDMS DML or CA IDMS DMLO command where an entity of the type represented by the variable is appropriate.

Variable entities are:

- **&R-RECORD** which is current of transaction.
- **&D-RECORD** most recently *displayed*; not necessarily the same as &R.
- **&A-AREA** associated with whatever record is mentioned in the CA IDMS DML command, or if no RECORD is mentioned, the AREA associated with current RECORD of transaction.
- **&S-SET** most recently mentioned in any CA IDMS DML command.
- **&T-SCRATCH AREA ID** most recently mentioned in any CA IDMS DML command.
- **&Q-QUEUE ID** most recently mentioned in any CA IDMS DML command.

The SHOW VARIABLES command may be used to display current values of &R, &D, &A, and &S.

## Macro Variables

The macro variables (&1... &9) give you the ability to replace variable symbols within a predefined (EQUATE or PF key) text string with tokens entered in the command line.

For example, assume the following EQUATE is in effect:

**EQUATE ONRW OBTAIN NEXT &1 WITHIN &2**

If you entered ONRW EMP-REC &A on the command line, the CA IDMS DML command actually executed would be:

**OBTAIN NEXT EMP-REC WITHIN &A**

Each &n encountered requires that there be a token on the command line available for substitution. The number itself is not really significant; i.e. the first macro variable in the command text string is replaced by the first available token, the next macro variable by the second available token, etc. Thus, the following two text strings would be processed exactly the same by CA IDMS DMLO:

**OBT NEXT &1 IN &2**

**OBT NEXT &7 IN &3**

**Note:** For more information on using macro variables, see [Extended Command Processing](#) (see page 111) and [Command Processing Examples](#) (see page 115).

## Positional Variables

Positional variables (*@1...@9*) give you the ability to replace variable symbols within a predefined (EQUATE or PF key) text string with tokens entered on the command line. Positional variables differ from macro variables in that the number (*@n*) is significant; i.e. *@2* always refers to the second available token, and *@4* always refers to the fourth. Also, because the number specifies exactly which token is referred to, that token can be used for replacement multiple times.

Suppose the following EQUATE is in effect:

```
EQUATE  ONRW  OBT NEXT @1 IN @2
```

This example behaves exactly as if the positional variables were macro variables:

Command line entry: ONRW EMP-REC &A

Executed command: OBTAIN NEXT EMP-REC WITHIN &A

Suppose the following EQUATE is in effect:

```
EQUATE  OLRW  OBT NEXT @1 WHERE @2 = @3 OR @2 = ZERO
```

This example illustrates the difference between positional and macro variables.

Command line entry: OLRW EMP-LR EMP-PAY 12.34

Executed command: OBTAIN NEXT EMP-REC WHERE EMP-PAY = 12.34 OR EMP-PAY = ZERO

Note the multiple use of *@2* (= EMP-PAY) in decoding the command.

**Note:** For more information on using positional variables, see [Extended Command Processing](#) (see page 111) and [Command Processing Examples](#) (see page 115).

## Selection Variables

You can use selection variables (?R, ?A, ?S, ?T, and ?) in any CA IDMS DML or CA IDMS DMLO command where an entity of the type represented by the variable is appropriate.

?R, ?A, ?S, and ?T display selection lists of subschema RECORDS, AREAS, SETS, and SCRATCH AREA ID's respectively. If an item is selected, that item will replace the selection variable in the command. The unqualified selection variable (?) displays a list of items based on context; e.g., if either SET or AREA is allowed by the CA IDMS DML syntax, "?" would cause a list of SETS to be presented first.

If no SET selection is made, a list of AREAS is presented.

Whenever a list is displayed for selection into an CA IDMS DML command, the command line is protected. Selection of an item "unprotects" the command line. To escape the selection process without making a selection, use the INTERRUPT key, or the PF key assigned to the END command.

The entity-specific selection variables (?R, ?A, ?S, and ?T) may also be entered as commands to provide lists of the entities.

For more information about examples of the Record, Area, Sets, and Scratch Display screens., see the [General Information](#) (see page 11).

## CLIST Variables

When executing a CLIST, you may specify up to 9 arguments which are used to replace CLIST variables imbedded in the CLIST source commands. The syntax of the EXEC command is:

```
EXEC [S.]clist-name arg1 arg2 arg3 . . . arg9
```

where the optional *arg1* through *arg9* will be used to replace CLIST variables %1 - %9 which may occur throughout the CLIST source.

Each *argn* may be any type of token acceptable to CA IDMS DMLO. Examples of acceptable tokens are names of RECORDS, SETS, AREAS, FIELDS, literals, figurative constants, keywords, arithmetic or logical operators. Each CLIST command is expanded by replacement of %*n* with the *argn* argument string.

CLIST variable symbols (%1-%9) may *not* be used within EQUATES, and should not be used in PF key- assigned text. These restrictions do not apply to other types of variables.

See [CLIST Examples](#) (see page 53) for CLIST examples using CLIST variables.

## Extended Command Processing

This section will give you the concepts used in constructing CA IDMS DMLO and CA IDMS DML commands, together with a comprehensive set of examples.

You can access detailed information concerning CA IDMS DMLO and CA IDMS DML command syntax using the online documentation.

Preparation of commands by the CA IDMS DMLO interpreter makes use of the following:

- Text keyed on the command line
- Text associated with PF/PA keys
- EQUATED phrases
- Variable symbol substitution

## Creating Executable Commands

Use these steps to create an executable command:

1. Determine "raw" command text
2. Expand all levels of EQUATE phrases
3. Substitute variable symbols

## Step 1: Determine "Raw" Command

There are 3 possible command entry combinations:

- **Command Line Entry WITHOUT PF Key Selection**

- ... Raw Command = Entered text from command line

- **PF/PA Key Selection WITHOUT Command Line Entry**

- ... Raw Command = Text associated with PF/PA key.

- **Command Line Entry AND PF Key Selection**

- ... IF first token in command line represents an CA IDMS DML or CA IDMS DMLO verb (either directly or as part of an equate phrase), assign command line text to PF key:

- Raw command = Entered text from command line.

- ... ELSE

- ... IF first token in command line is 'NULL', Clear any text associated with PF key

- ... ELSE

- ... Raw command = Concatenation of (PF text and command line text)

**Note:** This also provides one of the techniques for assigning (and clearing) text to a PF key; i.e., enter text on the command line and press the desired PF key. Another method of changing PF key associates is by using the SHOW PFKEYS command to access the PF Keys Display screen.

## Step 2: Expansion of All Levels of EQUATE Phrases

The raw command created above is scanned for any words which have been EQUATED to phrases. All such words are expanded. If any words within the expanded text present EQUATE phrases, they also are expanded -- until no further expansion is possible.



### Step 3: Substitution for Variable Symbols

The final step in preparing an executable command is the resolution of any symbols requiring variable substitution contained in the raw command text. Note that there are four categories of variable symbols—only two of which are resolved at this time:

- **Subschema Entity Variables** - **&R &A &S &D** These variables are resolved later, as the command is executed.
- **CLIST Variables** - **%1, %2, . . . %9** These variables were resolved earlier by the CLIST processor to create a text string similar to a command line entry.
- **MACRO Variables** - **&1, &2, . . . &9** These variables are resolved at this stage. See the discussion below.
- **POSITION Variables** - **@1, @2, . . . @9** These variables are resolved at this stage. See the discussion below.

CA IDMS DMLO replaces variable symbols by tokens which appear at the end of the raw command text. The number and placement of those tokens which are necessary to create a valid command depend on the type of variables which appear in the raw command.

### MACRO Variable Processing

CA IDMS DMLO processes MACRO variables such that the first "&" variable, regardless of its number, receives the first token available. The second "&" variable, again regardless of its number, receives the second available token. For example, CA IDMS DMLO processes the following raw commands identically:

```
OBT NEXT &1 IN &2 VENDOR VENDOR-AREA
OBT NEXT &4 IN &5 VENDOR VENDOR-AREA
```

The resultant command in both cases would be:

```
OBT NEXT VENDOR IN VENDOR-AREA
```

**Note:** Use of "&" variables precludes multiple use of a token in a command. That is, the appearance of &1 twice in a command is always regarded as a requirement for two different tokens to be substituted rather than for a single token in two places.

## Positional Variable Processing

Use of positional variables, on the other hand, provides a direct correlation between the number of the "@" variable, and the position of the token available of substitution. That is, the appearance of @4 within a raw command requires that there be at least four tokens for use in replacing "@" variables. Also, because the variable number and the position of the replacement are fixed, the same variable can be used more than once. Consider the following raw command example:

```
OBT FIR LR-VENDOR WHERE @1 = @2 OR @1 = @3 VENDOR-ID 123 456
```

The resultant command is:

```
OBT FIR LR-VENDOR WHERE VENDOR-ID = 123 OR VENDOR-ID = 456
```

## Combining "&" and "@" Variables

Both types of variable symbols can appear in the same raw command. In this case, tokens associated with "&" symbols must precede tokens associated with "@" symbols. For example:

```
OBT FIR &1 WHERE @1 = @2 or @1 = @3 LR-VENDOR VENDOR-ID 123 456
```

would be interpreted as follows:

```
OBT FIR LR-VENDOR WHERE VENDOR-ID = 123 OR VENDOR-ID = 456
```

## Commands Summary

To summarize, the final tokens of the raw command must comprise one replacement token for each "&" symbol which appears, followed by as many replacement tokens as the highest number associated with any "@" symbol appearing in the raw command text.

## Variable Symbol Restrictions

There are restrictions on the use of variable symbols:

- CLIST variable symbols may **NOT** be contained within EQUATE phrases. However, this does not preclude the use of CLIST variables as replacement tokens within the raw command.
- Replacement tokens (for MACRO and positional variables) must appear at the end of the raw command text. The variable symbols themselves may be buried arbitrarily deeply within a nest of EQUATE phrases.

## Command Processing Examples

This section provides a variety of command processing examples:

- **Example 1** - EQUATE, macro variables
- **Example 2** - EQUATE, macro variables, PF key assignment
- **Example 3** - EQUATE, macro variables, PF key usage
- **Example 4** - EQUATE, macro variables, PF key usage, command entry
- **Example 5** - EQUATE, macro variables, PF key usage, command entry
- **Example 6** - EQUATE, macro variables, PF key usage, command entry
- **Example 7** - Multiple EQUATEs, macro and positional variables, command entry

Examples 1-4 Assume That The Following Equate Is In Effect:

```
EQUATE ONRW OB NEXT &1 IN &2
```

### Example #1

*Command Line Entry*

```
ONRW VENDOR &A
```

*Key Pressed*

```
ENTER
```

*Result*

```
Execute
```

### Example #2

*Command Line Entry*

```
ONRW VENDOR &A
```

*Key Pressed*

```
PF5
```

*Result*

```
Assign ONRW VENDOR &A to PF5
```

```
Execute OBT NEXT VENDOR IN &A
```

**Note:** Because the phrase represented by ONRW begins with a verb (OBT), CA IDMS DMLO assigns the text to PF5 regardless of whether PF5 already had text associated with it.

**Example #3**

*Command Line Entry*

ONRW

*Key Pressed*

PF5

*Result*

Assign ONRW to PF5 (See note above). No execution of CA IDMS DML command (unresolved variables).

**Example #4**

*Command Line Entry*

VENDOR ?S

*Key Pressed*

PF5

*Result*

Create command ONRW VENDOR ?S (by concatenating PF text ahead of the command line text).

Execute OBT NEXT VENDOR IN ?S (results in SET selection list).

**Note:** Because the command line entry did *not* begin with a verb, no assignment of text to PF5 was made, but rather currently assigned text was used.

Examples 5-7 Assume That The Following Equates And Pf Key Assignments Are In Effect:

```
EQUATE ONWP OBT NEXT LR-WARE-PART
EQUATE OLRW OBT &1 LR-VEND-PART
EQUATE XWHE WHERE @1 = @2 OR @1 = @3
PF4 ONWP
PF5 OLRW
PF6 OLRW XWHE
```

**Example #5**

*Command Line Entry*

WHERE WAREHSE-NBR = '012'

*Key Pressed*

PF4

*Result*

Execute OBT NEXT LR-WAREHSE-PART WHERE WAREHSE-NBR = '012'

**Example #6***Command Line Entry*

FIRST

*Key Pressed*

PF5

*Result*

Execute OBT FIRST LR-VEND-PART

**Note:** The concatenation of PF text is ahead of command line text giving OLRW FIRST. All EQUATE phrases are expanded giving:

OBT &1 LR-VEND-PART FIRST

&1 <---(variable substitution)

After performing the substitution as indicated, the resultant command as shown above is executed.

**Example #7***Command Line Entry*

FIRST PART-SUFFIX 6 7

*Key Pressed*

PF6

*Result*

Execute

```
OBT FIRST LR-VEND-PART
WHERE PART-SUFFIX = 6 OR
PART-SUFFIX = 7
```

**Note:** The concatenation of PF text is ahead of command line text giving OLRW XWHE  
FIRST PART-SUFFIX 6 7.

ALL EQUATE phrases are expanded, giving:

```
OBT &1 LR-VEND-PART
WHERE @1 = @2 OR @1 = @3    FIRST    PART-SUFFIX    6    7
```

(variable substitution)----> @1 @1 @2 @3

After performing the substitution as indicated, the resultant command as shown above  
is executed.

# Chapter 4: Operation

---

This section contains the following topics:

[CA IDMS DMLO System Requirements](#) (see page 119)

[Online Documentation Print Utility](#) (see page 120)

[Installation](#) (see page 121)

[User Exit Module](#) (see page 122)

## CA IDMS DMLO System Requirements

### Terminal Type

- IBM3270-type terminal (models 2 through 5), including the 3279 color display

### Library Requirements

Library requirements depend on the operating environment under which CA IDMS DMLO is executing.

### Security Requirements

CA IDMS DMLO offers different levels of security. You decide which level of security checking you want CA IDMS DMLO to perform. Based on this decision, you may need to register user and subschema information in the CA IDMS dictionary.

## Online Documentation Print Utility

The Online Documentation Print Utility provided with CA IDMS DMLO allows error messages and other product information to be printed upon request.

The Target or Distribution source library member GSIPRINT (z/OS), TOOL JCL library member GSIPRINT.S (z/VSE), or the GSIPRINT EXEC (z/VM), downloaded from the CA IDMS DMLO installation media, contains the JCL to execute the Online Documentation Print Utility. The online documentation modules for CA IDMS DMLO processing are as follows.

The printed version of the online documentation is presented one screen per page and includes page reference indices for screen options. Characters highlighted in the online documentation appear bolded in the printed version.

**Note:** The characters "{ }", which are used to denote optional statements in online documentation modules, appear as "& &" when printed with the Online Documentation Print Utility. The character "|", used to denote "or" in online documentation modules, appears as ":" when printed with the Online Documentation Print Utility.

Module Name	Description
COMMANDS	CLIST Editing Commands
USDACMD	Product-specific editing commands
USDAKEY	Update PF Keys (CLIST Edit)
USDBIND	Bind Screen
USDCEDIT	CLIST Editing Screen
USDCLSEX	CLIST Examples
USDCMAIN	CLIST Maintenance Screen (selection list)
USDCOMEX	Command Processing Examples
USDDATA	Data Manipulation Screen
USDDML	DML Commands
USDDMLO	CA IDMS DMLO Commands
USDEQUAT	Show Equate Screen
USDFG55	New Features Part 1.
USDFG12	New Features Part 2.
USDINTER	Interrupt Screen
USDMABBR	Standard Abbreviations



---

<b>Module Name</b>	<b>Description</b>
USDMENU	Menu Mode Startup Screen
USDMSG	Error Messages
USDOPTN	SHOW Options/Environment Screen
USDPARAM	Save Session Parameters Screen
USDPFKEY	CA IDMS DMLO Mainline PF Key Assignment
USDPRMNT	PROFILE Maintenance Screen
USDRDYLS	Ready Mode Information Discussion
USDSELAR	Selection List - AREA
USDSELRE	Selection List - RECORD
USDSElse	Selection List - SET
USDSIGN	Signon Screen
USDTERM	Termination Status Screen
USDTUT00	Primary Menu
USDWHERE	Where Clause Syntax
USDX0001	Secondary Menu

---

## Installation

For more information on installing CA IDMS DMLO, see the appropriate *CA IDMS Installation and Maintenance Guide*.

## User Exit Module

CA IDMS DMLO gives you the ability to customize a user exit module which can be invoked after each DML command is executed. At installation time the dummy user exit module USDMLXIT is provided. This module simply returns control immediately to the DMLO session.

The calling of the user exit is controlled by the installation parameter option USEREXIT in the USDTPARM module.

- If USEREXIT=YES USDMLXIT is invoked after each DML command executed.
- If USEREXIT=NO USDMLXIT is never invoked.
- If USEREXIT=(DYNAMIC, ON) USDMLXIT is not invoked after each DML command is executed and is active for the session until the SET EXIT OFF command is executed.
- If USEREXIT=(DYNAMIC, OFF) USDMLXIT is not invoked but can be activated for the session with the SET ON command. This is the default setting.

**Note:** For more information, see the appropriate *CA IDMS Installation and Maintenance Guide*.

The user exit source member Assembler module USDMLXIT is delivered in source and can be customized by the user.

USDMLXIT is dynamically loaded at runtime and invoked by DMLO upon completion of all DML commands. This includes the SCRATCH and QUEUE commands. The module is delivered as a dummy module that simply returns control immediately to the DMLO session via a 'BR R14' instruction.

To customize the module, follow these rules:

1. Use standard IBM linkage conventions.
2. DSECT USDGLOB2 is used to pass information to and from USDMLXIT. Change only certain fields in this DSECT. If the restrictions are ignored, results are unpredictable and can cause subsequent DMLO errors. See source member USDGLOB2 for complete details.
3. Upon return from USDMLXIT to the DMLO session the USERCODE value in USDGLOB2 is processed as follows:
  - If USERCODE=1-9, DMLO terminates with the error message F8801 - F8809. Otherwise the DMLO session continues as normal.
  - If USERCODE is set by the user to a 1-9 value, DMLO processes the USERQUIT code as a completion indicator as follows:
    - 0 ROLLBACK
    - 1 FINISH

- If USERCODE=99, a 64-byte message field is passed back from the User Exit program. If the actual message length to be displayed is less than 64 bytes, then the 64-byte field must be padded with blank characters.
4. The customized source member USDMLXIT should be assembled and the link edited to create an executable load module named USDMLXIT. The link edit input statements to do this are as follows:

```
z/OS      INCLUDE OBJLIB(USDMLXIT)
          MODE AMODE(31), RMODE(ANY)
          ENTRY USDMLXIT
          NAME USDMLXI(R)
```

```
z/VSE     PHASE USDMLXIT,*
          MODE AMODE(31), RMODE(ANY)
          INCLUDE USDMLXIT
          ENTRY USDMLXIT
```

See source members USDMLXIT and USDGLOB2 for complete details on the User Exit functionality.



# Chapter 5: Messages

---

This section contains the following topics:

[Overview](#) (see page 125)

[Messages Generated by CA IDMS DMLO](#) (see page 125)

## Overview

This section lists all messages generated by CA IDMS DMLO. Included are the codes for messages, reasons for their occurrences, and suggestions for appropriate remedial actions.

## Messages Generated by CA IDMS DMLO

**CA IDMS DMLO messages** are in one of the following formats:

- *annnn* where *a* is a one-character identifier and *nnnn* is a unique message number
- *aaann* where *aaa* is a three-character identifier and *nn* is a unique message number

**EDITOR messages** are preceded by a unique eight-character code. The message code is in the format *USXnnns*, where *USX* is the identifier for the EDITOR, *x* is the identifier of the module issuing the message, *nnn* is the message number, and *s* is the severity code for the message. The severity code is either I, W, E, or F. Severity codes are explained below.

## Severity Codes

- **Informative**—A CA IDMS DMLO message beginning or ending with the letter I indicates an informative message. Informative messages need no remedial action.
- **Warning**—A CA IDMS DMLO message beginning or ending with the letter W indicates a warning. Warning messages report conflicting parameter data or processing conditions that may cause unexpected results.
- **Error**—A CA IDMS DMLO message beginning or ending with the letter E indicates an error. Error messages report erroneous and conflicting parameter data that has caused processing to terminate.
- **Fatal**—A CA IDMS DMLO message beginning or ending with the letter F indicates an error. Fatal messages require termination of the CA IDMS DMLO session.

**ELR31 LR SYNTAX ERROR**

**Reason:** Command syntax error encountered.

**Action:** Review the allowable syntax for the command and reenter.

**ELR32 LR SYNTAX ERROR**

**Reason:** Command syntax error encountered.

**Action:** Review the allowable syntax for the command and reenter.

**ELR33 LR SYNTAX ERROR**

**Reason:** Command syntax error encountered.

**Action:** Review the allowable syntax for the command and reenter.

**ELR34 LR SYNTAX ERROR**

**Reason:** Command syntax error encountered.

**Action:** Review the allowable syntax for the command and reenter

**ELR35 LR SYNTAX ERROR**

**Reason:** Command syntax error encountered.

**Action:** Review the allowable syntax for the command and reenter.

**ELR36 LR SYNTAX ERROR**

**Reason:** Command syntax error encountered.

**Action:** Review the allowable syntax for the command

**ELR37 VARIABLE SYMBOL/TOKEN MISMATCH**

**Reason:** The entered command requires variable substitution, but the command line contains insufficient tokens.

**Action:** Review the syntax and correct the command entry.

**ELR51 INVALID ENCODED VALUE**

**Reason:** An apparent internal error has occurred.

**Action:** If it recurs, contact Technical Support.

**ELR52 INVALID TOKEN**

**Reason:** An invalid token has been encountered during command parsing.

**Action:** Review the syntax and correct the command entry.

**ELR53 SUBSCRIPTING NOT IMPLEMENTED**

**Reason:** CA IDMS DMLO does not yet support entry of subscripted fields in this context.

**Action:** Reenter command without subscripted fields.

**ELR55 UNKNOWN QUALIFIER**

**Reason:** An unknown qualifier was encountered.

**Action:** Ensure the qualifier is required and is correct.

**ELR56 EXCESSIVE QUALIFIERS**

**Reason:** CA IDMS DMLO supports a maximum of four levels of qualification.

**Action:** Reenter the command with fewer qualifiers.

**ELR57 UNKNOWN ELEMENT NAME**

**Reason:** The indicated element is not known to CA IDMS DMLO.

Either the name is incorrect, or the record which contains the element has not yet been referenced.

**Action:** Correct the element name or ensure that the parent record is loaded (via the INIT command).

**ELR58 QUALIFICATION REQUIRED FOR -**

**Reason:** Multiple elements exist with the indicated name.

**Action:** Reenter the command with a different element or with qualifiers which allow CA IDMS DMLO to determine which particular element is wanted.

**ELR59 ELEMENT/QUALIFIER MISMATCH -**

**Reason:** The specified element does not correspond with the entered qualifier.

**Action:** Correct the element name or the qualifier.

**ELR5A AMBIGUOUS QUALIFICATION FOR -**

**Reason:** The qualification as entered is still insufficient to specify the element wanted.

**Action:** Change the qualification or add another level of qualification.

**ELR61 LR UNPAIRED PARENTHESIS**

**Reason:** Left and right parentheses must be paired.

**Action:** Review the syntax and correct the command entry.

**ELR62 LR LITERAL FORMAT ERROR**

**Reason:** A literal of incorrect format has been entered.

**Action:** Review the syntax and correct the command entry.

**ELR63 LR INTERNAL XDE ENCODING ERROR**

**Reason:** An apparent internal error has occurred.

**Action:** If it recurs, contact Technical Support.

**ELR64 LR INTERNAL ERROR - INVALID LIT FLAG**

**Reason:** An apparent internal error has occurred.

**Action:** If it recurs, contact Technical Support.

**ELR65 LR INTERNAL ERROR - DATA TYPE CONVERT**

**Reason:** An apparent internal error has occurred.

**Action:** If it recurs, contact Technical Support.

**ELR66 INVALID DATA IN FIELD -**

**Reason:** Data contained in the specified field does not conform to its usage, resulting in erroneous WHERE construct.

**Action:** Specify a different field or alter the data content of the field before entering the WHERE command.



**ESQ00 FUNCTION NOT IMPLEMENTED**

**Reason:** A scratch/queue function has been requested that is not yet implemented.

**Action:** Enter another command.

**ESQ01 INCOMPLETE COMMAND**

**Reason:** The scratch/queue command is incomplete.

**Action:** Check the syntax and reenter.

**ESQ02 INVALID KEYWORD**

**Reason:** The indicated word is inappropriate or misspelled.

**Action:** Check the syntax and reenter.

**ESQ03 RECORD/FIELD NEEDED AT**

**Reason:** The scratch/queue syntax requires a record or field name where indicated.

**Action:** Check the syntax and reenter.

**ESQ04 NUMERIC FIELD / LITERAL NEEDED AT**

**Reason:** The scratch/queue syntax requires a numeric field or literal where indicated.

**Action:** Check the syntax and reenter.

**ESQ05 INVALID LENGTH**

**Reason:** The specified length is negative or zero.

**Action:** Correct the contents of the specified length field or specify another field.

**ESQ06 FIELD NAME NEEDED AT**

**Reason:** The scratch/queue syntax requires a field name where indicated.

**Action:** Check the syntax and reenter.

**ESQ07 USAGE MUST BE BINARY FULLWORD**

**Reason:** The scratch/queue syntax requires that usage of the indicated field must be 4-byte binary.

**Action:** Specify a more appropriate field.

**ESQ08 IMPROPER OPTION**

**Reason:** The specified option is not appropriate for the scratch/queue command.

**Action:** Check the syntax and reenter.

**ESQ09 AMBIGUOUS KEYWORD**

**ESQ10 DUPLICATE OR CONFLICTING KEYBOARD**

**Reason:** The scratch/queue command contains a duplicate keyword or conflicting words.

**Action:** Check the syntax and reenter.

**ESQ11 NO ENTRIES TO DISPLAY**

**Reason:** A SHOW entity command was entered, and there are no occurrences of the entity to display.

**Action:** None.

**ESQ12 NO UNUSED DEFAULT BUFFERS**

**Reason:** The scratch/queue processor logic attempted to allocate an unused default buffer name to execute the command, but all 10 of the buffers have been used already.

**Action:** Specify an explicit buffer name in the command.

**ESQ13 UNABLE TO DETERMINE DATA AREA**

**Reason:** The scratch/queue processor is unable to determine a data buffer for a command.

**Action:** Specify an explicit buffer name in the command.

**ESQ14 UNABLE TO DETERMINE SCRATCH/QUEUE AREA ID**

**Reason:** The scratch/queue processor is unable to determine what area ID is wanted for a command. You may have included ?T in a SCRATCH command, but not selected an area from the resultant selection list.

**Action:** Explicitly specify a SCRATCH/QUEUE ID in the command.

**E1001 PRINT ERROR**

**Reason:** One of the following caused this message:

1. The DCB for the print file could not be opened.
2. For TSO users, this message may mean the TSO dynamic allocation of the SYSOUT dataset failed.

**Action:** Contact Technical Support.

**E2001 KEYWORD NOT RECOGNIZED - keyword/command/operand**

**Reason:** The keyword, command, and/or operand that was entered has not been defined to CA IDMS DMLO.

**Action:** There are three possible courses of action:

1. If the command has been equated, execute the SHOW EQUATE command to verify the 'new-symbol'.
2. See the appropriate section in this manual for the correct spelling.
3. Enter the corrected command operand, and/or keyword, then press the ENTER key.

**E2002 OPERAND MUST BE NUMERIC LITERAL**

**Reason:** A non-numeric operand was entered with the LINE command.

**Action:** Enter a numeric value and press the ENTER key.

**E2003 INVALID OPERAND**

**Reason:** Either of these reasons caused this message:

1. The record or the field name that was entered cannot be found.
2. The keyword 'WITHIN', was specified with the operand, 'SET *record-name*'.

**Action:**

1. If the first reason was the cause of the error, correct the record or the field name before reentering the command.
2. If the second reason was the cause of the error, delete the keyword 'WITHIN' before reentering the command.

**E2004 FIELD NOT UNIQUE - SPECIFY RECORD**

**Reason:** The field-name specified was found in multiple records.

**Action:** Reenter the command using the *record-name* operand to specify the record which contains the field you want to access.

**E2005 RECORD NAME NOT IN SUBSCHEMA**

**Reason:** The name of the record or field specified could not be found.

**Action:**

1. Verify the *record-name* and/or the *field-name*, or enter the 'SHOW RECORDNAMES' command. This will verify the existence of the record in the subschema that you are accessing.
2. Correct the operand that was in error; and reenter the command.

**E2006 FIELD NOT IN RECORD**

**Reason:** The *field-name* specified was not found in the record specified.

**Action:**

1. Verify the *field-name* and/or the *record-name*, and
2. Correct the operand that was in error; and reenter the command.

**E2010 RECORD NOT BOUND**

**Reason:** BIND has not been issued for the named record, and the AUTOBIND option is not active.

**Action:**

1. BIND the record -OR- SET AUTOBIND ON
2. Reenter the DISPLAY command.

**E2012 FIELD IS WITHIN REPEATING GROUP - DISPLAY RECORD**

**Reason:** The *field-name* specified is within a repeating group.

**Action:** To display this field, you must display the entire record and use the windowing commands. Follow these steps:

1. Enter the DISPLAY *record-name* command.
2. Enter the windowing commands (UP, DOWN, LEFT, RIGHT, LINE) to display the field you want.

#### **E2014 TWO OPERANDS REQUIRED**

**Reason:** Only one operand was found on the EQUATE command.

**Action:** There are two possible courses of action:

1. Check the screen display to make sure there is at least one space between operand 1 and 2.
2. Enter the second operand on the EQUATE command. See the CA IDMS DMLO online documentation for EQUATE details.

#### **E2015 SECOND OPERAND NOT DEFINED**

**Reason:** The second operand has not been defined to CA IDMS DMLO.

**Action:** Possible courses of action are as follows:

1. If the second operand is incorrect, reenter.
2. If the second operand is an element in a record which has not yet been referred to, enter CINIT *record-name* (to load elements from the dictionary). Retry the EQUATE command.
3. Use EQUATE to define the second operand as a synonym of a previously defined word. Then retry the original EQUATE command.

See the CA IDMS DMLO online documentation for command details.

#### **E2016 FIRST OPERAND ALREADY DEFINED**

**Reason:** The <new-symbol> entered on the EQUATE command has already been defined.

**Action:**

1. Enter the SHOW EQUATES command to display all active <new-symbols> and <old-symbols>.
2. Reenter the EQUATE command based on the information from the SHOW EQUATE command. (Refer to the CA IDMS DMLO online help facility for details on the EQUATE command.)

#### **E2017 OPERAND REQUIRED FOR SHOW**

**Reason:** The SHOW command was entered without an operand.

**Action:**

1. Refer to the CA IDMS DMLO online help facility for the valid operands for the SHOW command.
2. Reenter the command with at least one space before the operand.

#### **E2018 UNKNOWN SHOW REQUEST**

**Reason:** The operand specified in the SHOW command has not been defined to CA IDMS DMLO.

**Action:** There are two possible actions to be taken:

1. Validate the operand.
2. Enter the SHOW EQUATES command to determine if the operand has been given a new-symbol. (The <old-symbol> is still valid.)

#### **E2019 FORMAT ERROR(S)**

**Reason:** A format error(s) occurred when processing new or modified data was entered. The field entered was in the wrong format (for example, more than one decimal point entered in a numeric field. '1.2.3').

**Action:**

1. Enter the 'SET field-name NATIVE' command to determine the correct usage mode for the field.
2. Correct and reenter the data.

#### **E2020 INVALID COMMIT REQUEST**

**Reason:** The second operand entered in the COMMIT command is not defined to CA IDMS DMLO.

**Action:** Correct the second operand and reenter the command.

#### **E2021 INVALID ROLLBACK REQUEST**

**Reason:** The second operand of the ROLLBACK command has not been defined to CA IDMS DMLO.

**Action:** Correct the second operand and retry the command.

#### **E2022 SECOND COMMAND INVALID**

**Reason:** The first keyword or operand after the IF command is not defined to CA IDMS DMLO.

**Action:** Validate the keyword or operand and retry.

**E2023 USER ID MUST BE ENTERED**

**Reason:** The level of security which your installation has established requires your user ID to be entered before the signon process can be completed.

**Action:** Supply your assigned user ID. If you have not been assigned a user ID, see your DBA.

**E2024 PASSWORD IS REQUIRED**

**Reason:** Security established at your installation requires your password to be entered before the signon process can be completed.

**Action:** Key in your assigned password. If you have not been assigned a password, see your DBA.

**E2025 PLEASE ENTER SUBSCHEMA NAME**

**Reason:** The name of the subschema that you want to access must be supplied before you can signon to CA IDMS DMLO.

**Action:** Enter the subschema name through which you want to access the database.

**E2026 VERSION MUST BE NUMERIC**

**Reason:** The version number of the schema was entered as a non-numeric value.

**Action:** Reenter the version number using a numeric value.

**E2027 LOGICAL DATA WIDTH MUST BE NUMERIC**

**Reason:** The value that was entered for the logical data width was entered as a non-numeric value.

**Action:** Reenter the field as a numeric value. (This value must be from 80 to 264.)

**E2028 USER DOES NOT EXIST, TRY AGAIN**

**Reason:** The user ID entered has not been defined to CA IDMS DMLO.

**Action:** Validate the user ID. If it is valid, check with your DBA to verify that the user ID has been defined to CA IDMS DMLO.

#### **E2029 SUBSCHEMA/SCHEMA/VERSION COMBINATION NOT FOUND**

**Reason:** There are four possible causes:

1. If only the subschema was entered, this message indicates that the subschema entered could not be found in the directory.
2. If both subschema and schema were entered, this message indicates that the subschema is not associated with the schema entered.
3. If the subschema, schema, and version number was entered, this message means that this combination is not valid.
4. Subschema marked non-executable in dictionary.

**Action:** Verify the field(s) that were entered. If the fields are valid see your DBA.

1. If there is only one schema for the requested subschema, the schema is not required.
2. If you do not supply CA IDMS DMLO with schema and version information and duplicate subschema names are detected, and the subschemas are associated with a different version of the same schema, CA IDMS DMLO will select the subschema associated with the schema with the highest version number.

#### **E2030 PASSWORD INCORRECT - REENTER**

**Reason:** The password that was entered does not match the password that was registered for the user ID entered.

**Action:** Validate the password. If it is correct, validate the user ID. If the user ID is correct see your DBA.

#### **E2031 NOT AUTHORIZED FOR SUBSCHEMA ACCESS**

**Reason:** The user ID entered does not have authorization to access the requested subschema.

**Action:** Validate the user ID. If the user ID is correct see your DBA.

#### **E2032 MORE THAN ONE SCHEMA, PLEASE SPECIFY**

**Reason:** The subschema that was entered has multiple schemas defined.

**Action:** Enter the schema name for the subschema you want to access.



**E2033 INVALID DISPLAY OPTION**

**Reason:** The operand entered on the DISPLAY command has not been defined to CA IDMS DMLO.

**Action:** There are two possible courses of action:

1. Validate the operand on the DISPLAY command, or
2. If the operand has been EQUATED, execute the SHOW EQUATE command to verify the <new-symbol>.

Correct the operand and reenter the command.

**E2034 CONTINUE ONLY VALID FOR THE LAST**

**Reason:** The CONTINUE operand on the DISPLAY command can only be used if a second screen is waiting to be viewed. (See the CA IDMS DMLO online help facility for more details on the DISPLAY CONTINUE command.)

**Action:** Delete CONTINUE operand and reenter the command.

**E2035 NODE OR DICTIONARY COULD NOT BE BOUND**

**Reason:** Bind run-unit failed for node or dictionary specified.

**Action:**

1. Review status code in the command area to determine the cause.
2. If node or dictionary were specified incorrectly, type the correct node and/or dictionary name(s) and press the ENTER key.

**E2036 DBNAME GIVEN COULD NOT BE BOUND**

**Reason:** The subschema and DBNAME entered at signon could not be bound. (For more information, see the CA IDMS error status code 1469 information.)

**Action:**

1. Enter the correct spelling for DBNAME.
2. Check valid names for DBNAME options and try again.

**E2037 ONLY DBNAME CAN BE CHANGED**

**Reason:** More than DBNAME was changed for Error Message E2036 screen.

**Action:** Rekey only the new DBNAME and press the ENTER key, or press the PA1 key (you will get the Interrupt screen), key in FINISH and reinitialize your CA IDMS DMLO session.

**E2038 LOGICAL DATA WIDTH IS OUT OF RANGE**

**Reason:** The Logical Data Width entered is either less than 80 or greater than 264. A value outside of this range is not permitted.

**Action:** Enter a value between 80 and 264 as specified on the Signon screen.

**E2039 DDS TARGET NODE INACTIVE OR DISABLED**

**Reason:** The node name specified at signon is unavailable. (For more information, see CA IDMS Error Status Code 1480.)

**Action:** Choose another node or verify the node name you selected.

**E2040 SPECIFIED DATABASE UNKNOWN TO CV**

**Reason:** The DBNAME you supplied at signon was misspelled or inaccurate. (See CA IDMS Error Status Code 1481 for more information.)

**Action:** Verify the spelling of the DBNAME or check with your DBA on the spelling of names valid for your installation.

**E2041 SUBSCHEMA NOT ALLOWED UNDER SPECIFIED DATABASE**

**Reason:** The subschema name and DBNAME entered at signon are mutually exclusive. (See CA IDMS Error Status Code 1482 for more information.)

**Action:** Choose a different database and/or subschema. Check with your DBA for valid database and subschema combination(s).

**E2042 SET ORDER IS NOT SORTED**

**Reason:** A FORMAT 6 FIND/OBTAIN command named a set that is not a sorted set.

**Action:** Correct and reenter the command.

**E2043 RECORD DOES NOT PARTICIPATE IN ANY SORTED SETS**

**Reason:** A FORMAT 6 FIND/OBTAIN command named a record that is not a member of any sorted set.

**Action:** Correct and reenter the command.

**E2044 RECORD IS NOT A MEMBER OF SET**

**Reason:** A FORMAT 6 FIND/OBTAIN command named a record that is not a member of the sorted set named.

**Action:** Correct and reenter the command.

**E2045 PAGE NUMBER IS TOO LARGE**

**Reason:** A FORMAT 6 FIND/OBTAIN command specified a database key with a page number outside the range valid for your subschema.

**Action:** Correct the page number to be valid for your subschema.

**E2046 LINE INDEX NUMBER IS TOO LARGE**

**Reason:** A FORMAT 6 FIND/OBTAIN command specified a database key with a line index number outside the range valid for your subschema.

**Action:** Correct the line index number to be valid for your subschema.

**E2047 BIND OPERAND IS NOT A KEYWORD - CORRECT KEYWORD IS 'ALL'**

**Reason:** Allowable formats are BIND *record-name* or BIND ALL.

**Action:** Correct and reenter the command.

**E2048 INVALID OPERAND FOR BIND - VALID OPERAND IS 'ALL'**

**Reason:** Allowable formats are BIND *record-name* or BIND ALL.

**Action:** Correct and reenter the command.

**E2049 SUBSCHEMA MARKED NOT USABLE**

**Reason:** The subschema that was found had the ERR-026 flag set to 1.

**Action:** The subschema compile was not successful or a GENERATE was not issued. Correct the subschema or issue a GENERATE.

**E2050 PA KEY/PF KEY NOT RECOGNIZED**

**Reason:** The value entered on Signon screen for the INTERRUPT key is not one of the supported keys. Valid values are PA1, PA2, PA3, and PF1 through PF24.

**Action:** Change the value after the interrupt field on the Signon screen to one of the supported keys and reenter.

**E2051 KEYWORD MISSING**

**Reason:** At least one required keyword is missing from the entered command.

**Action:** Reference the command syntax and make the appropriate corrections.

**E4501 INVALID READY SYNTAX**

**Reason:** A CA IDMS DML READY command was entered using invalid syntax.

**Action:** Correct the syntax and reenter the command.

**E4502 READY USAGE-MODE SECURITY VIOLATION**

**Reason:** A ready request specified a secured usage mode. That is, the user is not authorized to ready the area in the usage mode requested.

**Action:** Contact your system administrator.

**E4505 AREA: area-name HAS ALREADY BEEN READIED**

**Reason:** A CA IDMS DML READY command was issued for an area that has already been readied. An area cannot be readied more than once. Also, an area's usage-mode cannot be changed during a session. Thus, attempting to re-ready an area in a different mode will not work.

**Action:** None needed; area-name is already readied. If an area's usage-mode must be changed, the current CA IDMS DMLO session must be terminated and a new one begun.

**E4506 RECORD NOT BOUND**

**Reason:** The *record-name* specified in an INITIALIZE command is not bound.

**Action:** Bind the record and reissue the command.

**E5001 DML OR DML/O COMMAND REQUIRED**

**Reason:** The first word on the command line is not recognized as a CA IDMS DML or CA IDMS DMLO command, either directly or as an EQUATE symbol.

**Action:** There are two possible courses of action:

1. Correct the spelling of the word.
2. Use the EQUATE command to establish a synonym.

**E5005 FIRST OPERAND NOT DEFINED**

**Reason:** The first operand is undefined in a command of the form EQUATE <symbol> NULL.

**Action:** Correct the first operand.

**E5006 FIRST OPERAND INVALID**

**Reason:** The first operand in an EQUATE begins with an invalid character.

**Action:** Correct the first operand.

**E5007 UNAVAILABLE VARIABLE**

**Reason:** Substitution was not possible for a variable symbol.

**Action:** If the symbol was &D, &R, &S, or &A, appropriate currency must be established.

If the symbol was a token variable (&1-&9), additional word(s) must be appended at the end of the command line.

**E5008 UNKNOWN VARIABLE**

**Reason:** A variable symbol other than &D, &R, &S, &D, &1 - &9 was found in a command.

**Action:** Correct the invalid symbol.

**E5009 UNAVAILABLE AND UNKNOWN VARIABLES**

**Reason:** Combination of E5007 and E5008.

**Action:** See above.

**E5010 INVALID DBKEY FORMAT**

**Reason:** The command contains an invalid database key literal.

**Action:** Correct the literal to an acceptable format:

ppppppp-llll X'hhhhhhhh' F'nnnnnnn'.

**E5011 SOURCE OPERAND INVALID**

**Reason:** Invalid source operand in MOVE command.

**Action:** See CLIST Example #2 for acceptable operands.

**E5012 KEYWORD "TO" MISSING**

**Reason:** Invalid syntax in MOVE command.

**Action:** See CLIST Example #2 for acceptable format.

**E5013 INCORRECT LENGTH FOR SOURCE FIELD**

**Reason:** MOVE source field must be four-byte field with binary or hex USAGE mode.

**Action:** Correct the command.

**E5014 OPTIONS MUST BE Y OR N**

**Reason:** Y/N are only valid selections to specify PROFILE data to be saved.

**Action:** Correct the selection.

**E5015 NO PROFILES FOR YOUR USER ID**

**Reason:** No PROFILES found for specified user ID for LOAD or ERASE.

**Action:** Change user ID or bypass PROFILE processing.

**E5016 PROFILE NAME MUST NOT BE BLANK**

**Reason:** Blank PROFILE name specified for PROFILE save function.

**Action:** Enter PROFILE name or bypass PROFILE processing.

**E5018 REQUESTED PROFILE NOT FOUND**

**Reason:** Specified PROFILE not found for LOAD or ERASE.

**Action:** Change PROFILE name or bypass PROFILE processing.

**E5019 ERROR RETRIEVING PAGE INFO**

**Reason:** A CA IDMS DMLO internal error occurred. The page info associated with the returned database key could not be retrieved. As a result, the database key could not be converted to decimal format. The database key displayed in hexadecimal format on the keypad display screen is accurate and set currency is updated.

**Action:** If the problem recurs, contact Technical Support.

**E5501 NO CLISTS AVAILABLE**

**Reason:** The user has requested either a CLIST or PROFILE list, but neither user nor global have been previously saved.

**Action:** Select another function.

**E5502 SOURCE USER ID NOT FOUND**

**Reason:** A COPY operation has been requested from another CA IDMS DMLO USER, but the user does not exist.

**Action:** Correct the user ID to one which exists, and retry the request.

**E5503 SOURCE CLIST NOT FOUND**

**Reason:** A COPY/RENAME operation has been requested, but the selected CLIST or PROFILE does not exist for the source user ID.

**Action:** Select another PROFILE or CLIST for the COPY operation.

**E5504 TARGET CLIST EXISTS**

**Reason:** An attempt has been made to overlay an existing CLIST or PROFILE with a COPY or RENAME operation.

**Action:** Either select another target CLIST or PROFILE name, or issue a REPLACE operation against the indicated CLIST or PROFILE.

**E5505 NO CLISTS FOR USER ID**

**Reason:** A maintenance operation has been specified, but no CLIST or PROFILE exists for the user.

**Action:** Either specify another user ID or select another function.

**E5506 REQUESTED CLIST NOT FOUND**

**Reason:** An operation against a specific CLIST or PROFILE member was attempted using a specific name, but the named CLIST or PROFILE does not exist.

**Action:** Either specify the correct CLIST or PROFILE name, or use the LIST function to specify the name choice.

**E5507 OPTIONS MUST BE Y OR N**

**Reason:** The specification of a PROFILE component for SAVE at the end of a session was specified as other than Y-Yes or N-No.

**Action:** Correct the indicated option and retry the transaction.

**E5508 PROFILE NAME MUST NOT BE BLANK**

**Reason:** You requested a PROFILE save at the end of a session, but did not supply a PROFILE name.

**Action:** Specify a PROFILE NAME and press the ENTER key, or press the indicated key to bypass the save operation.

**E5509 CLIST NOT FOUND**

**Reason:** You requested the execution of a CLIST, but the CLIST name you provided does not exist.

**Action:** Either specify a correct CLIST name, or use the LIST process to select a CLIST for execution.

**E5510 ONLY 9 ARGUMENTS ALLOWED**

**Reason:** A CLIST has been executed, but more than nine (9) argument variables were passed as part of the CLIST invocation.

**Action:** Correct the CLIST invocation stream to contain less than 9 variables.

**E5511 CLIST NAME REQUIRED**

**Reason:** An EXEC command was entered without a CLIST name.

**Action:** Supply a CLIST name and retry the transaction.

**E5512 COMMAND NOT ALLOWED AT THIS POINT**

**Reason:** You entered a command which is not allowed at this point in the CA IDMS DMLO session. For example you requested a PROFILE maintenance at other than signon time, or you entered a CLIST-only command when no CLIST is active.

**Action:** Remove the error command.



**E5513 UNAVAILABLE CLIST VARIABLE**

**Reason:** CA IDMS DMLO encountered a CLIST symbolic variable but you did not supply a value at CLIST you requested a PROFILE maintenance at other than signon time, or you entered a CLIST-only command when no CLIST is active.

**Action:** Alter the CLIST so that the variable is no longer required, or be sure to supply a value (or null indication ',') when required.

**E5514 CLIST RECURSION ERROR**

**Reason:** A CLIST has attempted to invoke another level of CLIST, but the lower-level CLIST is currently active at a higher level in the current CLIST nest.

**Action:** Remove the recursive CLIST call and reexecute.

**E5515 USER CLIST NAME REQUIRED**

**Reason:** The user has attempted to initiate an edit session on a CLIST, but did not supply a CLIST name.

**Action:** Supply a CLIST name or access the list process to select a CLIST.

**E5516 INTERNAL EDITOR ERROR**

**Reason:** An internal interface error has occurred when invoking an edit session on a CLIST.

**Action:** Contact Technical Support with this error and a description of the procedural steps that preceded this error.

**E5517 CLIST CURRENTLY IN USE**

**Reason:** You attempted to edit a CLIST, but the CLIST has been locked by another terminal or user.

**Action:** Specify another CLIST, or retry the request at a later time.

**E5518 SYSTEM CLIST MAY NOT BE CREATED**

**Reason:** You attempted to CREATE a CLIST for the SYSTEM level user ID, but you are not signed on as the SYSTEM user.

**Action:** Either CREATE the CLIST for a user other than the SYSTEM user, or signon as the SYSTEM user and CREATE the CLIST.

**E5519 INVALID COMMAND SYNTAX**

**Reason:** You entered invalid command syntax for a CLIST or PROFILE maintenance command.

**Action:** Correct the syntax.

**E5520 SYNTAX REQUIRES x**

**Reason:** CA IDMS DMLO encountered an improper Menu Mode. The parser presents the syntactical phrase which is required.

**Action:** Correct the syntax.

**E5521 ADDITIONAL OPERANDS NEEDED**

**Reason:** CA IDMS DMLO encountered an incomplete Menu Mode command.

**Action:** Correct the syntax.

**E5522 SUPERFLUOUS OPERANDS PRESENT**

**Reason:** CA IDMS DMLO encountered a Menu Mode command with too many operands.

**Action:** Correct the syntax.

**E5523 BIND OPERAND MUST BE RECORD-NAME OR "ALL"**

**Reason:** You entered a BIND command with a missing or improper operand.

**Action:** Correct the syntax.

**E5524 READY OPTION MUST BE R/U/P**

**Reason:** You entered a value other than R-Retrieval, U-Update, P-Profile, or blank in the initial CAIDMS DMLO Signon screen.

**Action:** Enter one of the indicated options, or enter a blank on the Signon screen and enter READYLIST during the session to explicitly READY areas.

**E5525 LOWER CASE AND OPTIONS 1-8 MUST BE Y/N**

**Reason:** You entered a value other than Y-Yes or N-No in one of the ENVIRONMENT options.

**Action:** Correct the indicated value to Y or N and retry the transaction.

**E5526 LOGICAL DATA WIDTH MUST BE NUMERIC AND 80-255**

**Reason:** You entered an invalid data width (screen width).

**Action:** Correct the data width and retry.

**E5527 INTERRUPT VALUE NOT RECOGNIZED**

**Reason:** You specified a value for the interrupt key which does not correspond to a PF key or PA key value.

**Action:** Correct the interrupt value and retry.

**E5528 INCOMPLETE COMMAND**

**Reason:** You entered an incomplete operand in a MOVE or CLIST IF statement.

**Action:** Correct the operand. Review online help or the *CA IDMS DML Online User Guide* for syntax information.

**E5529 INVALID LITERAL**

**Reason:** CA IDMS DMLO encountered an invalid literal in a MOVE or CLIST IF statement.

**Action:** Correct the syntax. Review online help or the *CA IDMS DML Online User Guide* for syntax information.

**E5530 "OF RECORD" NEEDED FOR DUP FIELD NAME**

**Reason:** A MOVE or CLIST IF statement contains an *element-name* which appears in more than one record in the current subschema.

**Action:** Specify OF RECORD-NAME to qualify the ELEMENT and retry the statement.

**E5531 FIELD/RECORD NOT FOUND**

**Reason:** An ELEMENT or RECORD in a MOVE or CLIST IF statement does not exist in the current dictionary/node.

**Action:** Correct the ELEMENT or RECORD name, and retry the statement.

**E5532 RELATIONAL NOT FOUND**

**Reason:** You specified an invalid relational operator in a CLIST IF statement.

**Action:** Correct the syntax.

**E5533 PREMATURE END OF IF GROUP**

**Reason:** CA IDMS DMLO reached the end of a CLIST, but an ENDIF was not encountered for an active CLIST IF statement.

**Action:** Correct the IF statement and retry the CLIST.

**E5534 MISSING KEYWORD 'TO'**

**Reason:** A MOVE statement does not contain the TO keyword.

**Action:** Correct the MOVE statement and retry the transaction.

**E5535 INVALID SOURCE FIELD DATA**

**Reason:** CA IDMS DMLO encountered an inconsistent SOURCE operand in a MOVE statement.

**Action:** Correct the MOVE statement and retry the transaction.

**E5546 INVALID KEYWORD**

**Reason:** The IFC or MOVE command contains an invalid keyword.

**Action:** Review the syntax for the command of interest, and reenter.

**E5547 NEED RECORD NAME, FOUND**

**Reason:** The syntax requires a *record-name* where the indicated token was found.

**Action:** Reenter the command with a *record-name*.

**E5548 SUBSCRIPTING NOT IMPLEMENTED**

**Reason:** Reference to a subscripted field has appeared in the IFC/MOVE command. This is currently unsupported by CA IDMS DMLO.

**Action:** Remove the offending reference.

**E5550 MEMBER RECORD NOT IN SUBSCHEMA**

**Reason:** The specified record is not included in the current subschema.

**Action:** Specify a different *record-name*, or start a new CA IDMS DMLO session with a different subschema.

**E6001 REC NOT IN SUBSCHEMA,VERSION NOT IDD-OWNED**

**Reason:** *Record-name* not member of current subschema, or the specified record/version is not in the dictionary.

**Action:** Change *record-name* or version number.

**E6002 NO ELEMENTS EXIST FOR RECORD**

**Reason:** Record has no associated elements.

**Action:** Enter new *record-name*, or do a dictionary update for the named record.

**E6003 INTERNAL HASH ERROR**

**Reason:** An internal error occurred in the CA IDMS DMLO hash lookup routine.

**Action:** If the problem recurs, contact Technical Support.

**E6008 RECORD NOT IN SPECIFIED AREA**

**Reason:** A FIND/OBTAIN specifying both record and area was entered for which the record does not reside in the specified area.

**Action:** Change the record or area as required, or use &A to ensure the area matches the record.

**E6009 AMBIGUOUS KEYWORD**

**Reason:** The syntax of the current command allows words which are not distinguishable based on the number of letters entered for the specified word.

**Action:** Reenter the command using more leading letters for the specified keyword. You should refer to the command syntax to determine the minimum spelling.

**E6012 VERSION MISMATCH WITHOUT REPLACE OPTION**

**Reason:** The INIT command has been specified for a dictionary-owned record/version that has already been loaded from the dictionary. The specified version does not match the prior version.

**Action:** To load a new version of the same record, you must specify the REPL option on the INIT command.

**E6013 VERSION REQUIRED FOR REPLACE OPTION**

**Reason:** The REPL option of the INIT command has been specified without the VERSION option.

**Action:** Add a VERSION specification to the command.

**E6016 UNAUTHORIZED REQUEST**

**Reason:** The specified maintenance command is restricted for use by your CA IDMS DMLO system administrator.

**Action:** Contact your CA IDMS DMLO system administrator.

**E6017 NO ROWS TO PRINT**

**Reason:** A PRINT request was specified, but there are no display rows formatted.

**Action:** Issue a DISPLAY command, then a PRINT command.

**E6019 MEMBER RECORD NOT BOUND**

**Reason:** An attempt was made to construct a search key for an OBTAIN/RETURN with USING option, and the record that is the member of the sorted set has not been bound and AUTOBIND is not active.

**Action:** Set AUTOBIND ON or INIT *record-name* and reissue the OBTAIN/RETURN command.

**E6020 INVALID TARGET DATA TYPE**

**Reason:** You entered a command that specified a literal as the target item of a MOVE statement.

**Action:** Change the target data item to a record or field name and reissue the command.

**F0000 NORMAL TERMINATION**

**Reason:** CA IDMS DMLO terminated normally.

**Action:** Press the ENTER key to complete the exit from CA IDMS DMLO.

**F0003 TERMINATED BY FINISH OR ROLLBACK DURING STARTUP**

**Reason:** The PA1 key was pressed during the signon process and either a FINISH or ROLLBACK command was entered on the returned screen. An error status of 0069 is also returned with this message. This is because the run-unit was never bound.

**Action:** Press the ENTER key to complete the exit from CA IDMS DMLO.

**F0069 SUBSCHEMA CONTROL TABLE NOT FOUND**

**Reason:** The address of the subschema control table cannot be found. This is because the address was never established or it was once established, but now it is lost.

**Action:** Reinvoke CA IDMS DMLO, and try again. If the problem recurs, contact Technical Support.

**F0077 RUN-UNIT COULD NOT BE FOUND**

**Reason:** The subschema entered on the signon screen was located, but it could not be loaded. This message is for CA IDMS/DC environments only.

**Action:** The subschema you are accessing must be on either the core image library if you are running under z/VSE, or if you are running under z/OS, or z/VM, it must be in the load library.

**F0100 TERMINAL READ/WRITE ERROR**

**Reason:** The terminal read or write routine has issued a return code greater than zero to CA IDMS DMLO. A possible line or terminal error has occurred.

**Action:** Reinvoke CA IDMS DMLO, and try again. If the problem recurs, contact Technical Support.

**F0102 INTERNAL SCAN ERROR ON INPUT DATA**

**Reason:** The 3270 scan routine cannot locate the end-of-record marker for the data string that was entered.

**Action:** Reinvoke CA IDMS DMLO, and try again. If the problem recurs, contact Technical Support.

**F0103 TEMPORARY STORAGE ACQUIRE ERROR**

**Reason:** One of the following caused this message:

1. A non-zero return code was returned from CICS/VS temporary storage, or
2. An ID error was encountered.

**Action:** Reinvoke CA IDMS DMLO, and try again. If the problem recurs, contact Technical Support.

**F0105 NO RECORDS TO BIND**

**Reason:** CA IDMS DMLO could not find any records in the subschema that was being accessed.

**Action:** Verify the subschema. If records do exist, contact Technical Support.

**F0106 BIND RUN UNIT FAILED FOR CA IDMS DMLO**

**Reason:** The bind run unit failed for the subschema you requested.

**Action:**

1. Correct the cause for the CA IDMS status code. The code is displayed on the screen.
2. Reinvoke CA IDMS DMLO.

**F0107 DUPLICATE KEYWORD FOUND**

**Reason:** CA IDMS DMLO located a duplicate keyword in the keyword table.

**Action:** Contact Technical Support.

**F0108 DUPLICATE HASH ENTRY**

**Reason:** CA IDMS DMLO located a duplicate area or record name in the internal table.

**Action:** Contact Technical Support.

**F0109 NO AREAS TO READY**

**Reason:** CA IDMS DMLO could not find any area in the subschema being accessed.

**Action:** Verify the subschema. If areas do exist, contact Technical Support.



**F0118 RECORD HAS NO FIELDS**

**Reason:** A request was entered to process a record that did not have any elements defined.

**Action:** Verify the record description in the subschema. If elements do exist, contact Technical Support.

**F1002 PUSH STACK OVERFLOW**

**Reason:** CA IDMS DMLO received a non-zero return code from the storage routine.

**Action:** Allocate more storage to CA IDMS DMLO. If the problem recurs, contact Technical Support.

**F1003 ENTER STACK OVERFLOW**

**Reason:** CA IDMS DMLO received a non-zero return code from the storage routine.

**Action:** Allocate more storage to CA IDMS DMLO. If the problem recurs, contact Technical Support.

**F1718 INVALID USAGE FOR DEPENDS ON FIELD**

**Reason:** A record with an occurs depending on group has an object field whose usage mode is not numeric.

**Action:** Verify the record. If the field is defined as numeric, contact Technical Support.

**F1719 INVALID USAGE FOR SUBSCRIPTED FIELD**

**Reason:** The usage mode for a subscripted field must be filler, hexadecimal, or picture X.

**Action:** Contact Technical Support.

**F1938 PACKED FIELD GREATER THAN 16 BYTES**

**Reason:** CA IDMS DMLO encountered a field with a usage mode of packed decimal and a length greater than 16 bytes.

**Action:** Contact Technical Support.

**F2001 BIND RUN UNIT FAILED FOR DIRECTORY**

**Reason:** The bind run unit failed for subschema 'IDMSNWKA'.

**Action:**

1. Correct the cause for the CA IDMS status code. The code is displayed on the screen.
2. Reinvoke CA IDMS DMLO.

**F2002 BIND RECORD FAILED**

**Reason:** A request to bind a record in the subschema 'IDMSNWKA' failed.

**Action:**

1. Correct the cause for the CA IDMS status code. The code is displayed on the screen.
2. Reinvoke CA IDMS DMLO.

**F2003 READY AREA FAILED**

**Reason:** A request to ready area 'DDLDML' failed.

**Action:**

1. Correct the cause for the CA IDMS status code. The code is displayed on the screen.
2. Reinvoke CA IDMS DMLO.

**F2004 OBTAIN FOR SSA-024 FAILED**

**Reason:** A non-zero status was returned while CA IDMS accessed the dictionary record 'SSA-024' that relates to your subschema.

**Action:**

1. Correct the cause for the CA IDMS status code. The code is displayed on the screen.
2. Reinvoke CA IDMS DMLO.

**F2005 OBTAIN FOR SS-026 FAILED**

**Reason:** A non-zero status was returned while CA IDMS was accessing the dictionary record 'SS-026' that relates to your subschema.

**Action:**

1. Correct the cause for the CA IDMS status code. The code is displayed on the screen.
2. Reinvoke CA IDMS DMLO.

**F2006 OBTAIN FOR SSR-032 FAILED**

**Reason:** A non-zero status was returned while CA IDMS was accessing the dictionary record 'SSR-032' that relates to your subschema.

**Action:**

1. Correct the cause for the CA IDMS status code. The code is displayed on the screen.
2. Reinvoke CA IDMS DMLO.

**F2007 OBTAIN FOR SSOR-034 FAILED**

**Reason:** A non-zero status was returned while CA IDMS was accessing the dictionary record 'SSOR-034' that relates to your subschema.

**Action:**

1. Correct the cause for the CA IDMS status code. The code is displayed on the screen.
2. Reinvoke CA IDMS DMLO.

**F2008 OBTAIN FOR SR-036 FAILED**

**Reason:** A non-zero status was returned while CA IDMS was accessing the dictionary record 'SR-036' that relates to your subschema.

**Action:**

1. Correct the cause for the CA IDMS status code. The code is displayed on the screen.
2. Reinvoke CA IDMS DMLO.

**F2009 OBTAIN FOR SDR-042 FAILED**

**Reason:** A non-zero status was returned while CA IDMS was accessing the dictionary record 'SDR-042' that relates to your subschema.

**Action:**

1. Correct the cause for the CA IDMS status code. The code is displayed on the screen.
2. Reinvoke CA IDMS DMLO.

**F2010 OBTAIN FOR ACCESS-045 FAILED**

**Reason:** A non-zero status was returned while CA IDMS was accessing the dictionary record 'ACCESS-045' that relates to your subschema.

**Action:**

1. Correct the cause for the CA IDMS status code. The code is displayed on the screen.
2. Reinvoke CA IDMS DMLO.

**F2011 OBTAIN FOR USER-047 FAILED**

**Reason:** A non-zero status was returned while CA IDMS was accessing the dictionary record 'USER-047' that relates to your subschema.

**Action:**

1. Correct the cause for the CA IDMS status code. The code is displayed on the screen.
2. Reinvoke CA IDMS DMLO.

**F2012 OBTAIN FOR PROG-051 FAILED**

**Reason:** A non-zero status was returned while CA IDMS was accessing the dictionary record 'PROG-051' that relates to your subschema.

**Action:**

1. Correct the cause for the CA IDMS status code. The code is displayed on the screen.
2. Reinvoke CA IDMS DMLO.

**F2013 OBTAIN FOR SSSDR-074 FAILED**

**Reason:** A non-zero status was returned while CA IDMS was accessing the dictionary record 'SSDR-074' that relates to your subschema.

**Action:**

1. Correct the cause for the CA IDMS status code. The code is displayed on the screen.
2. Reinvoke CA IDMS DMLO.

**F2014 OBTAIN FOR RCDSYN-079 FAILED**

**Reason:** A non-zero status was returned while CA IDMS was accessing the dictionary record 'RCDSYN-079' that relates to your subschema.

**Action:**

1. Correct the cause for the CA IDMS status code. The code is displayed on the screen.
2. Reinvoke CA IDMS DMLO.

**F2015 OBTAIN FOR NAMESYN-083 FAILED**

**Reason:** A non-zero status was returned while CA IDMS was accessing the dictionary record 'NAMESYN-083' that relates to your subschema.

**Action:**

1. Correct the cause for the CA IDMS status code. The code is displayed on the screen.
2. Reinvoke CA IDMS DMLO.

**F2016 OBTAIN FOR SRCD-113 FAILED**

**Reason:** A non-zero status was returned while CA IDMS was accessing the dictionary record 'SRCD-113' that relates to your subschema.

**Action:**

1. Correct the cause for the CA IDMS status code. The code is displayed on the screen.
2. Reinvoke CA IDMS DMLO.

**F2017 CALC-FIELD NAME NOT FOUND**

**Reason:** Your subschema contains a partial view of a record whose location mode is CALC. The CALC keyfield is not contained with the partial view.

**Action:** CA IDMS DMLO will not process this type of situation. Reinvoke CA IDMS DMLO and specify a different subschema.

**F2018 FINISH FOR DIRECTORY FAILED**

**Reason:** A non-zero status was returned from either the FINISH for the run unit using the subschema 'IDMSNWKA', or from the Directory Interface.

**Action:**

1. Correct the cause for the CA IDMS status code. The code is displayed on the screen.
2. Reinvoke CA IDMS DMLO.

**F2019 DUPLICATE RECORD NAME ENCOUNTERED**

**Reason:** Duplicate record names have been found in the requested subschema. CA IDMS DMLO cannot process subschemas that have duplicate record names.

**Action:** Validate your subschema. If no duplicate record names exist, contact Technical Support.

**F2020 INVALID FIELD BLOCK CHAIN POINTER**

**Reason:** An error has occurred within CA IDMS DMLO.

**Action:** Contact Technical Support.

**F2021 DUPLICATE SET NAME ENCOUNTERED**

**Reason:** Duplicate set names have been found in the requested subschema. CA IDMS DMLO cannot process subschemas that have duplicate set names.

**Action:** Validate your subschema. If no duplicate set names exist, contact Technical Support.

**F2022 DUPLICATE AREA NAME ENCOUNTERED**

**Reason:** Duplicate area names have been found in the requested subschema. CA IDMS DMLO cannot process subschemas that have duplicate area names.

**Action:** Validate your subschema. If no duplicate area names exist, contact Technical Support.

**F2023 INVALID PICTURE FOR FIELD**

**Reason:** The field name and picture clause which follow the message are invalid or not supported with the current release.

**Action:** If the picture is invalid, correct the source.

**F2024 BIT FIELD LENGTH TOO LONG FOR DISPLAY**

**Reason:** The requested bit field has exceeded your logical data width display area. (You defined this area when you signed on to CA IDMS DMLO. For more information on logical data width refer to the CA IDMS DMLO online help facility.)

**Action:**

1. Reinvoke CA IDMS DMLO.
2. Supply CA IDMS DMLO with a number in the logical data width field to accommodate the length of the bit field you need to display.

**F2025 SSR-032 DLENGTH DATA INCORRECT (SHORT)**

**Reason:** The record length defined in the SSR-032 record is less than the actual record length because of an invalid record length calculation for records containing BIT fields.

**Action:** Reinvoke CA IDMS DMLO and bypass binding any records with BIT fields. CA IDMS DMLO cannot access these records.

**F2026 SESSION STORAGE UNAVAILABLE**

**Reason:** Request for storage returned a 'NOT AVAILABLE' condition.

1. The subschema being accessed requires more session storage than was available.
2. System load may be too high.
3. Dynamic storage pool defined in the TP environment may be too small.

**Action:**

1. Use a smaller subschema.
2. Wait until the system load decreases.
3. Increase TP environment dynamic storage pool. (For example, in CA IDMS/DC, increase STORAGE POOL SIZE; in CICS, increase DYNAMIC STORAGE POOL.)

**F2027 PROGRAM LOAD FAILURE**

**Reason:** A module required by CA IDMS DMLO could not be loaded.

**Action:** Verify that USDMAIN, USDDRCT, USDERRS are located in the library used for the CA IDMS DMLO environment.

**F2028 OBTAIN FOR SSMR-068 FAILED**

**Reason:** A non-zero status was returned while CA IDMS was accessing the dictionary record 'SSMR-068' that relates to your subschema.

**Action:**

1. Correct the cause for the CA IDMS status code. The code is displayed on the screen.
2. Reinvoke CA IDMS DMLO.

**F2029 OBTAIN FOR SSCR-070 FAILED**

**Reason:** A non-zero status was returned while CA IDMS was accessing the dictionary record 'SSCR-070' that relates to your subschema.

**Action:**

1. Correct the cause for the CA IDMS status code. The code is displayed on the screen.
2. Reinvoke CA IDMS DMLO.

**F2030 SORTED SET IS MISSING SSCR-070**

**Reason:** A sorted set is indicated by your subschema but no SSCR-070 records were found.

**Action:**

1. Correct the cause for the missing SSCR-070 record(s).
2. Reinvoke CA IDMS DMLO.

**F2031 INTERNAL ERROR, MISSING SET HTE**

**Reason:** CA IDMS DMLO could not find a set control block.

**Action:** Contact Technical Support.

**F2033 DUPLICATE RECORDS FOUND FOR PROGRAM DBMSDMLO**

**Reason:** More than one record for the program DBMSDMLO was found in the data dictionary. Only one such record may be present.

**Action:** Contact your system administrator.

**F2034 INVALID VERSION NUMBER FOR PROGRAM DBMSDMLO**

**Reason:** The version number parameter supplied to program DBMSDMLO must have the value 1 or 2 or 3. It does not.

**Action:** Contact your system administrator. The version number supplied to program DBMSDMLO must match the requested security level.

**F2100 IDMSNWKA BIND FAILURE IN SUBSCH-CHK**

**Reason:** Attempt to BIND dictionary subschema failed in USDSSLDO.

**Action:** Have DBA check CA IDMS and CA IDMS DMLO environment.



**F2101 IDMSNWKA BIND FAILURE IN FIELD-LOAD**

**Reason:** Attempt to BIND dictionary subschema failed in USDSSLDO.

**Action:** Have DBA check CA IDMS and CA IDMS DMLO environment.

**F2110 IDD RECORD BIND FAILURE IN SUBSCH-CHK**

**Reason:** Attempt to BIND dictionary subschema failed in USDSSLDO.

**Action:** Have DBA check CA IDMS and CA IDMS DMLO environment.

**F2111 IDD RECORD BIND FAILURE IN FIELD-LOAD**

**Reason:** Attempt to BIND a dictionary network record failed in USDSSLDO.

**Action:** Have DBA check CA IDMS and CA IDMS DMLO environment.

**F2120 DDLDDL READY FAILURE IN SUBSCH-CHK**

**Reason:** Attempt to READY dictionary area DDLDDL failed in USDSSLDO.

**Action:** Have DBA check CA IDMS and CA IDMS DMLO environment.

**F2121 DDLDDL READY FAILURE IN FIELD-LOAD**

**Reason:** Attempt to READY dictionary area DDLDDL failed in USDSSLDO.

**Action:** Have DBA check CA IDMS and CA IDMS DMLO environment.

**F4001 BIND RUN UNIT FAILED FOR USDPROF**

**Reason:** PROFILE module USDPROF<sub>x</sub> could not BIND.

**Action:** Ensure that all installation instructions for the PROFILE module and database have been followed.

**F4002 BIND RECORD FAILED FOR USDPROF**

**Reason:** PROFILE module USDPROF<sub>x</sub> could not BIND.

**Action:** Ensure that all installation instructions for the PROFILE module and database have been followed.

**F4003 READY AREA FAILED FOR USDPROF**

**Reason:** PROFILE module USDPROF<sub>x</sub> could not READY USD-DATA-AREA.

**Action:** Ensure that all installation instructions for the PROFILE module and database have been followed.

**F4020 GETSTG REQUEST FAILED IN USDPROF**

**Reason:** USDPROF<sub>x</sub> was unable to obtain storage.

**Action:** If problem occurs repeatedly, the storage pool in your TP environment may need to be enlarged.

**F4100 USDPROF IDMS-STATUS ERROR**

**Reason:** Improper error-status from PROFILE database i/o.

**Action:** Ensure that all installation instructions for the PROFILE module and database have been followed.

**F4440 CA IDMS/DMLO INTERNAL ERROR**

**Reason:** A CA IDMS DMLO internal error occurred.

**Action:** If problem recurs, contact Technical Support.

**F4441 CA IDMS/DMLO INTERNAL ERROR**

**Reason:** A CA IDMS DMLO internal error occurred.

**Action:** If problem recurs, contact Technical Support.

**F4442 CA IDMS/DMLO INTERNAL ERROR**

**Reason:** A CA IDMS DMLO internal error occurred.

**Action:** If problem recurs, contact Technical Support.

**F4443 CA IDMS/DMLO INTERNAL ERROR**

**Reason:** A CA IDMS DMLO internal error occurred.

**Action:** If problem recurs, contact Technical Support.

**F4444 CA IDMS/DMLO INTERNAL ERROR**

**Reason:** A CA IDMS DMLO internal error occurred.

**Action:** If problem recurs, contact Technical Support.

**F4445 CA IDMS/DMLO INTERNAL ERROR**

**Reason:** A CA IDMS DMLO internal error occurred.

**Action:** If problem recurs, contact Technical Support.

**F4446 CA IDMS/DMLO INTERNAL ERROR**

**Reason:** A CA IDMS DMLO internal error occurred.

**Action:** If problem recurs, contact Technical Support.

**F4447 CA IDMS/DMLO INTERNAL ERROR**

**Reason:** A CA IDMS DMLO internal error occurred.

**Action:** If problem recurs, contact Technical Support.

**F4448 CA IDMS/DMLO INTERNAL ERROR**

**Reason:** A CA IDMS DMLO internal error occurred.

**Action:** If problem recurs, contact Technical Support.

**F4449 CA IDMS/DMLO INTERNAL ERROR**

**Reason:** A CA IDMS DMLO internal error occurred.

**Action:** If problem recurs, contact Technical Support.

**F5001 FREEMAIN FAILURE**

**Reason:** A CA IDMS DMLO internal error occurred.

**Action:** If problem recurs, contact Technical Support.

**F5101 ATTEMPT BEYOND START OF TERMINATOR STACK**

**Reason:** A CA IDMS DMLO internal error occurred.

**Action:** If problem recurs, contact Technical Support.

**F7701 LINK FAILED**

**Reason:** LINK to PROFILE module, USDPROF, failed.

**Action:** Ensure that USDPROF<sub>x</sub> has been linked and defined to your TP monitor according to installation instructions.

**F7702 LINK FOR GSIHELP FAILED**

**Reason:** LINK to HELP module, GSIHELP, failed.

**Action:** Ensure that the installation instructions appropriate for your TP monitor have been followed.

**F7708 LINK FOR USXUMAN FAILED**

**Reason:** Access to full-screen editor failed.

**Action:** Ensure that CA IDMS DMLO and other Computer Ensure that CA IDMS DMLO and other CA Products have been correctly installed, and the libraries are accessible to CA IDMS/DC.

**F7709 LINK FOR USXDRVR FAILED**

**Reason:** Access to full-screen editor failed.

**Action:** CA products have been correctly installed, and the libraries are accessible to CA IDMS/DC.

**F7720 LOAD FOR USDPROF0 FAILED**

**Reason:** Access to CA IDMS DMLO profile module failed.

**Action:** Ensure that CA IDMS DMLO and other CA Products have been correctly installed, and the libraries are accessible to CA IDMS/DC.

**F7721 LOAD FOR USDCMNT0 FAILED**

**Reason:** Access to CA IDMS DMLO CLIST module failed.

**Action:** Ensure that CA IDMS DMLO and other CA Products have been correctly installed, and the libraries are accessible to CA IDMS/DC.

**F8801 USDMLXIT(01)**

**F8802 USDMLXIT(02)**

**F8803 USDMLXIT(03)**

**F8804 USDMLXIT(04)**

**F8805 USDMLXIT(05)**

**Reason:** Requested termination message from your installation's USDMLXIT user exit program.

**Action:** See your DBA for specifics.

**F8806 USDMLXIT(06)**

**F8807 USDMLXIT(07)**

**F8808 USDMLXIT(08)**

**F8809 USDMLXIT(09)**

**Reason:** Requested termination message from your installation's USDMLXIT user exit program.

**Action:** See your DBA for specifics.

**F9000 SECURITY VIOLATION - TOO MANY ATTEMPTS**

**Reason:** The user ID or the password was entered incorrectly three times.

**Action:** Re-invoke CA IDMS DMLO and enter the correct user ID and/or password.

**F9800 INTERNAL ERROR IN HASH LOGIC**

**Reason:** CA IDMS DMLO internal error.

**Action:** Contact Technical Support.

**F9900 USDTPARM - MISSING INITIAL EYECATCH**

**Reason:** The customization module USDTPARM does not have the expected format.

**Action:** Review the installation procedure with particular attention to changes to, and assembly of USDTPARM.

**F9901 USDTPARM - CORRUPT MENU-MODE COMMAND TABLE**

**Reason:** The customization module USDTPARM does not have the expected format.

**Action:** Review the installation procedure with particular attention to changes to, and assembly of USDTPARM.

**F9902 USDTPARM - CORRUPT MENU-MODE STATIC LINES**

**Reason:** The customization module USDTPARM does not have the expected format.

**Action:** Review the installation procedure with particular attention to changes to, and assembly of USDTPARM.

**F9903 USDTPARM - CORRUPT MENU-MODE HELP TEXT**

**Reason:** The customization module USDTPARM does not have the expected format.

**Action:** Review the installation procedure with particular attention to changes to, and assembly of USDTPARM.

**F9904 USDTPARM - CORRUPT SS EXCLUSION LIST**

**Reason:** The customization module USDTPARM does not have the expected format.

**Action:** Review the installation procedure with particular attention to changes to, and assembly of USDTPARM.

**F9905 USDTPARM - CORRUPT DISPLAY CHARACTERS**

**Reason:** The customization module USDTPARM does not have the expected format.

**Action:** Review the installation procedure with particular attention to changes to, and assembly of USDTPARM.

**F9906 USDTPARM - CORRUPT KEYWORD TBL BOUNDS**

**Reason:** The customization module USDTPARM does not have the expected format.

**Action:** Review the installation procedure with particular attention to changes to, and assembly of USDTPARM.

**F9907 USDTPARM - CORRUPT KEYWORD TBL ENTRY**

**Reason:** The customization module USDTPARM does not have the expected format.

**Action:** Review the installation procedure with particular attention to changes to, and assembly of USDTPARM.

**F9990 INTERNAL ERROR**

**Reason:** CA IDMS DMLO internal error.

**Action:** Contact Technical Support.

**F9991 INTERNAL ERROR**

**Reason:** CA IDMS DMLO internal error.

**Action:** Contact Technical Support.

**F9992 INTERNAL ERROR - MAIN(XPNDVAR)**

**Reason:** CA IDMS DMLO internal error.

**Action:** Contact Technical Support.

**F9993 VB50 EYECATCHER MISSING**

**Reason:** Subschema control blocks not as expected.

**Action:** Contact Technical Support.

**F9994 VB50 EYECATCHER MISSING**

**Reason:** Subschema control blocks not as expected.

**Action:** Contact Technical Support.

**F9995 INTERNAL ERROR**

**Reason:** CA IDMS DMLO internal error.

**Action:** Contact Technical Support.

**F9996 INTERNAL ERROR**

**Reason:** CA IDMS DMLO internal error.

**Action:** Contact Technical Support.

**F9997 INTERNAL ERROR**

**Reason:** CA IDMS DMLO internal error.

**Action:** Contact Technical Support.

**F9998 INTERNAL ERROR**

**Reason:** CA IDMS DMLO internal error.

**Action:** Contact Technical Support.

**F9999 INTERNAL ERROR VALUE TOO LARGE**

**Reason:** An error has occurred within CA IDMS DMLO.

**Action:** Contact Technical Support.

**ISQ01 ACTION = (K)ill ACTION = (K)ill, (S)elect -or- Pxx to exit**

**Reason:** SHOW SCRATCH command entered, or ?T variable appeared in command.

**Action:** In either case you may DELETE (K) a scratch area. If ?T was used, you must select an item, or press the indicated key to return to processing.

**I1001 PRINT SUCCESSFUL**

**Reason:** The print that was requested has been completed.

**Action:** None. This message is informative only.

**I2001 FIELD RESET TO USAGE MODE LENGTH nnn BYTES,**

**Reason:** The field-name was previously changed to hexadecimal usage mode. This message is returned after the field has been reset to its native mode by using the 'SET field-name NATIVE' command.

**Action:** None. This message is informative only.

**I2002 USAGE CHANGED**

**Reason:** The field or record in the 'SET field-name HEX' command was changed to hexadecimal usage mode.

**Action:** None. This message is informative only.

**I2003 EQUATE SUCCESSFUL**

**Reason:** This message is displayed after a successful execution of an EQUATE command.

**Action:** None. This message is informative only.



**I2004 NO PF KEYS ARE ASSIGNED YET**

**Reason:** This message indicates there have not been any PF keys programmed during this CA IDMS DMLO session.

**Action:** None. This message is informative only.

**I2005 FALSE**

**Reason:** This message is displayed as the result of a false condition in the CA IDMS DML 'IF' command.

**Action:** None. This message is informative only.

**I2006 TRUE**

**Reason:** This message is displayed as the result of a true condition in the CA IDMS DML 'IF' command.

**Action:** None. This message is informative only.

**I2007 CHECK ALL RECORDS TO BE BOUND**

**Reason:** This message is displayed after successful completion of the initialization process. CA IDMS DMLO is prompting you to indicate which records you want access to (or bound) during the CA IDMS DMLO session.

**Action:** To bind a record, position the cursor in the unprotected area next to the record to be bound and enter any keyboard character, then press the ENTER key. To bind all records, key in BIND ALL in the command input area, and press the ENTER key.

**I2008 INDICATE READY MODE FOR AREAS - SU SR PU PR EU ER**

**Reason:** This message appears when the READYLIST command is entered and the READY SCREEN appears. CA IDMS DMLO is prompting you to enter usage modes for the areas that are to be readied.

**Action:** To enter the usage mode for an area, position the cursor in the unprotected area following the area name; enter one of the modes indicated in the message; and press the ENTER key. (Refer to the appropriate CA IDMS reference manual for more information on usage modes.)

**I2009 NOTHING TO SHOW FOR REQUEST**

**Reason:** The SHOW command was executed. The requested operand did not have any data to display.

**Action:** None. This message is informative only.

**I4504 ALL AREAS READIED IN (SHARED) RETRIEVAL**

**Reason:** A CA IDMS DML READY command was issued with no operands. By default, this results in all areas being readied in a usage mode of shared retrieval. This message is a reminder that this has occurred.

**Action:** None needed unless some or all areas should have been readied in a different usage mode. If this is the case, terminate the CA IDMS DMLO session by entering a FINISH command, and begin a new session. (Once an area has been readied, its usage mode cannot be changed.) In the new session, ready areas by using individual CA IDMS DML READY commands, or use the CA IDMS DMLO READYLIST feature.

**I5001 AUTOHEX MODE NOW on/off**

**Reason:** The SET command for this option was processed.

**Action:** None.

**I5002 COBOL DISPLAY MODE NOW on/off**

**Reason:** The SET command for this option was processed.

**Action:** None.

**I5003 COMMAND DISPLAY WILL BE as input / as used**

**Reason:** The SET command for this option was processed.

**Action:** None.

**I5004 LOWER CASE OPTION NOW on/off**

**Reason:** The SET command for this option was processed.

**Action:** None.

**I5005 EXIT FOR ALL VERBS NOW on/off**

**Reason:** The SET EXIT .. ALL was processed.

**Action:** None.

**I5006 EXIT FOR DML VERB NOW on/off**

**Reason:** The SET EXIT .. verb was processed.

**Action:** None.

**I5007 EQUATE CANCELED**

**Reason:** An EQUATE symbol NULL has been processed.

**Action:** None.

**I5008 PF KEY COMMAND CLEARED**

**Reason:** A NULL command has been "autoset" to a PF key.

**Action:** None.

**I5009 MACRO COMMAND ACCEPTED**

**Reason:** A macro command has been "autoset" to a PF key.

**Action:** None.

**I5010 PROFILE RESTORED**

**Reason:** A PROFILE has been restored.

**Action:** See below.

**I5011 MAKE ANY REQUIRED CHANGES AND/OR PRESS ENTER**

**Reason:** A PROFILE has been restored. Any items may be changed.

**Action:** Press the ENTER key when variables are as needed.

**I5012 ERROR-STATUS 0000 RETURNED FROM finish / rollback**

**Reason:** A successful FINISH or ROLLBACK was processed.

**Action:** None.

**I5013 NO VARIABLES CURRENTLY SET**

**Reason:** A SHOW VARIABLES request was entered, but none of the variables have values.

**Action:** None.

**I5014 PROFILE xxxxxxxx ERASED**

**Reason:** A ( PROFILE %ERASE ) request was processed.

**Action:** None.

**I5015 ALL PROFILES ERASED FOR USER**

**Reason:** A ( %ALL %ERASE ) request was processed.

**Action:** None.

**I5536 EXEC CLIST in STEP MODE**

**Reason:** Indicates the operation mode under which the CLIST is operating.

**Action:** None.

**I5537 MAPIN DATA PROCESSED IN STEP MODE**

**Reason:** Indicates the operation mode under which MAPIN data is processed.

**Action:** None.

**I5538 AUTOBIND OF RECORDS IS NOW ON**

**Reason:** Indicates the status of the SET AUTOBIND option.

**Action:** None.

**I5539 CLIST XXXXXXXX ERASED**

**Reason:** The specified CLIST or PROFILE has been erased.

**Action:** None.

**I5540 ALL CLISTS ERASED FOR USER**

**Reason:** All CLISTS or PROFILES have been erased for the current user.

**Action:** None.

**I5541 COPY/REPLACE COMPLETED**

**Reason:** The current maintenance operation was successfully completed.

**Action:** None.

**I5542 RENAME COMPLETED**

**Reason:** The specified CLIST or PROFILE has been successfully renamed.

**Action:** None.

**I5543 SELECTED CLISTS ERASED**

**Reason:** The specified CLISTS or PROFILES have been successfully erased.

**Action:** None.

**I5544 PROMPT : AWAITING REPLY**

**Reason:** CA IDMS DMLO encountered a "PROMPT" command in the current CLIST.

**Action:** Execution will continue after you enter the RESUME command. You may enter any valid commands prior to that time.

**I6004 RECORD FOUND : VERSION**

**Reason:** The dictionary-owned record has been found with the indicated version number.

**Action:** None.

**I6005 ACTION = (B)ind, (I)nit ACTION = (B)ind, (I)nit, (S)elect -or- Pxxx to exit**

**Reason:** A SHOW RECORDS command was entered, or ?R was encountered in a command.

**Action:** No action required for SHOW RECORDS command. If ?R is part of a command, however, you must either (S)elect an entry or press the indicated key to continue processing.

You may enter B or I in either case.

**I6006 ACTION = READY mode ACTION = READY mode, (S)elect -or- Pxxx to exit**

**Reason:** A SHOW AREAS command was entered, or ?A was encountered in a command.

**Action:** No action required for SHOW AREAS command. If ?A is part of a command, however, you must either (S)elect an entry or press the indicated key to continue processing.

You may enter a READY mode for any non-readied area in either case.

**I6007 ACTION = any character to Select**

**Reason:** A ?S selection variable for SET was encountered in a command.

**Action:** You must (S)elect an entry or press the indicated key to continue processing.

**I6014 INVALID CHARACTER TRANSLATE RESULT IS NOW x**

**Reason:** The command "SET INVCHAR x" was entered.

**Action:** None.

**I6015 ACTION = (S)elect or PAX to exit**

**Reason:** A list of available subschemas has been requested.

**Action:** Select a subschema or press the indicated key to return to the Signon screen.

**I6018 nnnn PROFILES/CLISTS KILLED**

**Reason:** A "mass delete" request has been processed.

**Action:** None; this message is for information only.

**USXD002E INVALID COMMAND**

**Reason:** An invalid primary command was encountered. The cursor will be on the command.

**Action:** Check the syntax of the command. Correct and reenter the command.

**USXD202E INVALID PARAMETER ON COMMAND**

**Reason:** A valid command has been entered with an unrecognizable parameter.

**Action:** Check the syntax of command parameter. Correct and reenter the parameter.

**USXD204E ENTER LOCATE PARAMETER**

**Reason:** A line number was not included as a parameter in the LOCATE command.

**Action:** Check the syntax of the command. Correct the command (include line number parameter) and reexecute.

**USXD206E ONLY A NUMBER, MAX, HALF, CSR, OR PAGE IS ALLOWED**

**Reason:** An illegal option has been used with the LEFT, RIGHT, UP, or DOWN command.

**Action:** Check the syntax of the parameter. Correct the parameter (you may have included some erroneous characters in the syntax line) and reexecute.

**USXD208E ONLY "ON" OR "OFF" IS ALLOWED. "ON" IS THE DEFAULT**

**Reason:** An illegal option has been used with the NULLS command or CAPS command.

**Action:** Check the syntax of the parameter. Correct the parameter (you may have included some erroneous characters in the syntax line) and reexecute.

**USXD210E NO PARAMETER IS ALLOWED ON COMMAND**

**Reason:** A parameter has been included in the syntax of the CANCEL command which excludes parameters.

**Action:** Check the syntax of the command. Correct the command (delete the parameter) and reexecute.

**USXF001I FOUND CHARS "string"**

**Reason:** The string indicated has been found in the text. The cursor will be positioned at the beginning of the found string.

**Action:** None.

**USXF003I REP "string" WITH "replacement-string"**

**Reason:** The string indicated has been replaced with the replacement string. The cursor will be positioned at the replacement string.

**Action:** None.

**USXF004E MUST ENTER A FIND OR CHANGE COMMAND FIRST**

**Reason:** The RFIND (repeat find) or RCHANGE (repeat change) command was entered without a FIND or a CHANGE command being entered first.

**Action:** Be sure to enter the FIND command or the CHANGE command before the RFIND command or the RCHANGE command.

**USXF005I BOTTOM OF DATA REACHED**

**Reason:** The bottom of the text was reached before the string was found. If you enter an RFIND (repeat find) or an RCHANGE (repeat change) at this time, the editor will begin searching for the string at the top of the text.

**Action:** None.

#### **USXF006E ENCLOSE STRING IN QUOTES**

**Reason:** A single quotation mark was found where a pair is required or embedded blanks were found in the search string.

**Action:** Check quotation marks, correct, and reenter the command. If a string has embedded blanks, enclose it in quotation marks.

#### **USXF007E INVALID FIND SYNTAX**

**Reason:** The string specified in the FIND command is invalid. If the string has embedded blanks or quotation marks, it may appear to the editor to be more than one string; you can only specify one string in the FIND command.

**Action:** Check the syntax of the command. If the string has embedded blanks, enclose it in quotation marks. If the string has embedded single quotation marks, enclose it in double quotation marks. If the string has embedded double quotation marks, enclose it in single quotation marks. Then reenter the command.

#### **USXF008E INVALID CHANGE SYNTAX**

**Reason:** A string specified in the CHANGE command is invalid. If one of the strings has embedded blanks or quotation marks, it may appear to the editor that you are entering more than two strings; you can only specify two strings in the CHANGE command.

**Action:** Check the syntax of the command. If a string has embedded blanks, enclose it in quotation marks. If a string has embedded single quotation marks, enclose it in double quotation marks. If a string has embedded double quotation marks, enclose it in single quotation marks. Then reenter the command.

#### **USXF009I NO CHAR "string" FOUND**

**Reason:** The editor has searched the entire text without finding the indicated string.

**Action:** None.

#### **USXF010W ERROR CHARACTERS "string"**

**Reason:** The string used to replace the one found is too long.

**Action:** Adjust the length of the found string or reduce the length of the replacement string to match the length of the one found.



**USXF012I FOUND nnn OCCURRENCES OF "string"**

**Reason:** A string specified in the FIND ALL command has been found the specified number of times.

**Action:** None.

**USXF014I CHANGED nnn OCCURRENCES OF "string"**

**Reason:** A string specified in the CHANGE ALL command has been changed in the text the specified number of times.

**Action:** None.

**USXF016E CHANGE COMMAND NOT ENABLED IN BROWSE MODE**

**Reason:** The CHANGE command was used in the Browse Option to search for and change the next occurrence of a string.

**Action:** None. The CHANGE command can only be used when editing text (Edit Option) and not while in Browse mode.

**USXF018W CHANGED nnn OCCURRENCES WITH nnn ERRORS OF "string"**

**Reason:** The string identified in the CHANGE ALL command has been changed successfully (nnn OCCURRENCES) with (nnn ERRORS).

**Action:** Look for the error message on the line number field(s) and make the necessary correction(s).

**USXI002E INVALID COMMAND**

**Reason:** An invalid line command has been entered.

**Action:** Check the syntax of the line command on the line number field. Correct and reenter the command.

**USXI006E INVALID ON THIS LINE**

**Reason:** A line command has been entered on a line for which the command is invalid.

**Action:** Reenter the line command on an appropriate line.

**USXI008E INVALID SCROLL AMOUNT**

**Reason:** An invalid Scroll Option was entered in the SCROLL field.

**Action:** Check the syntax of the Scroll Option next to the word SCROLL. Be sure the option starts in the left-most position of the field. Correct syntax or position.

**USXL002W BLOCK COMMAND INCOMPLETE**

**Reason:** A partial sequence of Line Commands, in block form, is incomplete and cannot be executed.

**Action:** If you are just scrolling or using the FIND command to locate the line to complete the block command, this message serves as a warning. Complete the command and execute. Otherwise, check the syntax of the commands to be sure all commands in block form are in pairs. Correct the commands and reexecute.

**USXL004E COMMAND CONFLICT**

**Reason:** A sequence of Line Commands cannot be executed.

**Action:** Check the order of the commands. Correct the commands and reexecute.

**USXL006W MOVE/COPY PENDING**

**Reason:** A move or copy cannot be executed because of a missing 'A'(after), 'B' (before), 'C' (copy), or 'M' (move) line command or COPY Primary command.

**Action:** Enter the missing command.

**USXL008E ITERATION COUNT MISMATCH**

**Reason:** Block (RR) line command iterations have conflicting values.

**Action:** Check the syntax of the command. Make sure that the values of the iterations match (one or both may be left blank).

**USXL009E LINE COMMAND RANGE CONFLICT**

**Reason:** A sequence of Line Commands cannot be executed because the range of one line command overlaps the range of another.

**Action:** Check the order of the commands. Correct the commands and reexecute.

**W2001 MODIFIED DATA ENTERED, COMMAND IGNORED**

**Reason:** This message is displayed when data has been entered in the unprotected area and in the command input area. The data entered in the unprotected area always takes precedence over any command entered. (Refer to the CA IDMS DMLO online help for a detailed discussion.)

**Action:** To enter the command, rekey the first character in the displayed command and press the ENTER key.

**Note:** This will not occur if MAPIN mode is FAST.

**W2003 OCCURS DEPENDING ON CONTROL FIELD ALTERED - REDISPLAY RECORD**

**Reason:** A control field in an 'OCCURS DEPENDING ON' clause has been changed.

**Action:** CA IDMS DMLO has changed the field in the record data buffer. If you want to add or delete data in a field, enter 'DISPLAY record-name'. This will reformat the record to allow data to be added or deleted in the display record buffer area.

**W2004 A REDEFINING FIELD WAS MODIFIED**

**Reason:** This message indicates that a field in a redefined group has been changed.

**Action:** CA IDMS DMLO has changed the field in the record data buffer. To view the record in its new format enter the 'DISPLAY record-name' command. This will avoid possible confusion when the display is different.

**W2008 RECORD DISPLAY INCOMPLETE - TOO MANY (FIELDS/OCCURS)**

**Reason:** The record to be displayed could not fit into the display buffer area. The display buffer area ran out of line storage space.

**Action:** Enter the 'DISPLAY record-name CONTINUE' command to view the remaining fields.

**W2009 RECORD DISPLAY INCOMPLETE - FIELD TOO LONG**

**Reason:** The record to be displayed could not fit into the display buffer area. The display buffer area ran out of line storage space.

**Action:** Enter the 'DISPLAY record-name CONTINUE' command to view the remaining fields. (If you were displaying a field, the entire record must be displayed to use the CONTINUE option.)

**W2010 USAGE MODE MAY NOT BE CHANGED FOR A BIT FIELD**

**Reason:** The usage mode can be changed for any type of field except bit.

**Action:** None.

**W4503 READY AREA(S) FAILED...STAT: nnnn AREA: area-name**

**Reason:** An attempt to ready an area failed. If an unqualified ready was attempted, one of the readies failed, and others in the group may not have been processed. In the message, area-name is the name of the area that could not be readied; nnnn is the CA IDMS status code generated when the failure occurred.

**Action:** Correct the situation described by the status code, and reenter the READY command, or contact your system administrator.

**W5001 MODIFIED DATA ENTERED, PF KEY IGNORED**

**Reason:** Record data was changed at the same time that a PF key associated with a command was pressed, and MAPIN mode was not FAST.

**Action:** None.

**W5002 EXIT CHANGES DISALLOWED**

**Reason:** Installation parameters disallow changes to user exit specifications.

**Action:** None.

**W5003 HELP NOT YET AVAILABLE FOR TSO**

**Reason:** HELP function requested but not available for this environment.

**Action:** None.

**W5004 HELP NOT YET AVAILABLE FOR CICS**

**Reason:** HELP function requested but not available for this environment.

**Action:** None.

**W5545 RECORD ALREADY BOUND**

**Reason:** Record has already been bound.

**Action:** None.

# Appendix A: CLIST Editing Commands

---

This section contains the following topics:

[Overview](#) (see page 181)

[Editing Commands](#) (see page 181)

[Primary Commands](#) (see page 185)

[Line Commands](#) (see page 199)

[Text Manipulation Line Commands](#) (see page 204)

## Overview

This appendix provides a guide to the CLIST editing commands that are available in the CA IDMS DML Online CLIST editor. This appendix describes each command and its syntax.

**Note:** If you are not the CA IDMS DMLO system administrator, an attempt to edit a global CLIST forces you into a browse rather than an edit session. You may, however, copy global CLISTs into one of your own CLISTs during an edit session.

## Editing Commands

There are the following types of editing commands:

- Scroll Options
- Primary Commands
- Line Commands

## Scroll Options

Scroll options are used to determine how many lines or columns of the module to scroll up, down, right, or left when using a primary command or a PF key.

## Primary Commands

Primary commands are used to:

- Locate the desired line of a module
- Find the next occurrence of a string
- Reset the screen display to remove all line commands, column markers, and extraneous messages
- Cancel changes made with the editor to a module
- Turn the CAPS Mode on or off
- Display the time and date
- Copy source lines from one CLIST module to another
- Terminate Editor session, save changes made to the CLIST, and return to CA IDMS DMLO session mainline
- Save changes made to a CLIST and remain in Editor session

## Line Commands

Line commands are used to:

- Copy source lines within the module
- Move source lines within the module
- Specify the location at which source lines are to be copied or moved
- Repeat source lines in the module
- Delete source lines
- Insert blank source lines
- Display a line with column markings across the screen

## Program Function Keys

PF keys are set to many frequently used commands. This allows you to enter a command from any position on the Edit screen with one keystroke. In addition, the PA1, PA2, and CLEAR keys are set to redisplay the screen. Use the KEYS primary command to change Editor PF key settings.

To execute a single command set for a PF, press that key. The command executes when you press the PF key.

## Entering Commands

The following are descriptions of where commands are entered:

- **Primary Commands**—Enter these commands at the left side of the second line, after the word **COMMAND**. This field is called the **COMMAND** line.
- **Scroll Options**—Enter these options at the far right side of the second line on the screen, after the word **SCROLL**.
- **Line Commands**—Enter these commands in the line number fields at the left of the source.

Example	Function
SUSpend	Keywords appear in mixed case. The minimum required portion of each keyword appears in uppercase.
Find string	Variables appear in lowercase. You substitute an appropriate value for each variable.
Up [number-of-lines]	Brackets indicate optional clauses or commands.
	Braces enclose two or more options. Select an option.
	<pre>       r/      \     ] CAPS  &lt; ON &gt;             L\ OFF /     ]           </pre>
	A left arrow indicates the default value.
	<pre>       r/      \     ] CAPS  &lt; ON◀ &gt;             L\ OFF /     ]           </pre>

Item	Meaning
Order of Commands	You must enter a B (before) or an A (after) line command in conjunction with the COPY primary command and the C (copy) and M (move) line commands to indicate where to copy or move the lines.

Item	Meaning
Entering Blanks In Commands	Blanks (character spaces) are ignored in line command sequences, so you can enter blanks between a command and a value without affecting processing.  You must enter at least one blank (character space) between a primary command and a primary command value. You cannot embed blanks in a keyword.
Command Stacking	You can enter multiple primary and line commands. Primary commands must be separated by a semicolon (;).

## Scroll Options

```

/ Page          \
< Half         >
| Csr          |
\ number-of-lines /
    
```

where:

**Page**-specifies that a whole screen is to be scrolled whenever an UP, DOWN, RIGHT, or LEFT command is used.

**Half**- specifies that a half screen is to be scrolled whenever an UP, DOWN, RIGHT, or LEFT command is used.

**Csr**-specifies that the line with the cursor on it is to become: the bottom line displayed whenever the UP command is used, the top line whenever the DOWN command is used, the left-most column whenever the RIGHT command is used, or the right-most column whenever the LEFT command is used.

**number-of-lines**-specifies that this number of lines are to be scrolled whenever an UP, DOWN, RIGHT, or LEFT command is used.

Scroll options are used to specify how much of the screen is scrolled when you use an UP, DOWN, RIGHT, or LEFT primary command (or corresponding PF key) by itself.

At the far-right side of the second line on the Edit screen, the word **SCROLL** appears followed by one of the scroll options. To change the current setting, enter one of the other options over the current setting. The scroll option you set will remain in effect until you enter a different setting.



## Primary Commands

Primary commands are entered on the second line of the Edit screen after the word **COMMAND**. You can enter more than one primary command at a time. Use the following syntax:

**command;command**

### BOTTOM Command

#### **BOTtom**

The **BOTTOM** command displays the last full screen at the bottom of the module.

This command is available in Edit and Browse Modes.

### BOUNDS Command

#### **BOUnds**

The **BOUNDS** command sets the left and right column bounds. These bounds are saved in the edit profile. In addition, the **BOUNDS** command alters the action of the **FIND**, **CHANGE**, **Line Shifts**, and other commands that are column-sensitive.

You must specify left and right bounds, or neither, and the left bound must be smaller than the right bound. If bounds are specified incorrectly or without an operand, the default is the dataset minimum.

This command is available in Edit Mode.

### CANCEL Command

#### **CANceL**

Use the **CANCEL** command to cancel all changes made to the module since the last **SAVE** and to exit the Edit screen. You are returned to the previous display.

This command is available in Edit Mode.

## CAPS Command

```

    r/ ON ◀ \r
CAPS |◀ OFF ▶|
    \l      /l
  
```

The CAPS command is used to turn the CAPS Mode on and off. With the CAPS Mode on, all new alpha data is translated into uppercase. With the CAPS Mode off, the data remains unaffected. Data that was initially entered with the CAPS Mode off will remain in lowercase unless you edit the field.

This command is available in Edit and Browse Modes.

## CHANGE Command

```

    r/      \r r/      \r r/      \r r/      \r
CHANGE |◀ ALL ▶| |◀ string ▶| |◀ replacement-string ▶| | [Lb[rB]] |◀ X ▶|
    \l FIRST /l \l *      /l \l *      /l \l NX      /l
  
```

where:

**string**-specifies the string of characters to find and replace by *replacement-string*.

**replacement-string**-specifies the string of characters used to replace *string*.

**\*** (**asterisk**)-specifies the string value from the last FIND or CHANGE command entered.

**ALL**-specifies that all occurrences of a string are to be replaced in scanned lines.

**lb rb**-specifies left and right bounds (column positions) for the find. If specifying just the left bound, the string can be found anywhere within those bounds.

**X**-specifies only excluded lines are to be scanned.

**NX**-specifies only non-excluded lines are to be scanned.

Use the CHANGE command to search for and change the next occurrence of a string in the module. The Editor begins searching at the position of the cursor when you enter the command, and it searches downward until the string is found. If the cursor is on the COMMAND line when you enter the command, the editor begins searching at the top line displayed. If the string is not found, it is changed to the replacement string.

This command is available in Edit Mode.

## Change Command Rules

- ALL, FIRST, and the 'lbrb' can appear in any order, but the *replacement-string* must follow *string*.
- If a string has embedded blanks, enclose the string in either single or double quotes. For example:  

```
CHANGE 'program nmae' 'program name'
```

```
CHANGE "program nmae" "program name"
```
- If a string has a single asterisk (\*), number, ALL, or FIRST, enclose *string* in quotation marks:  

```
CHANGE '*' 'comments'
```
- If a string has leading quotation mark (single or double), enclose the string in quotation marks of the opposite kind. For example:  

```
CHANGE "'t' t"
```
- If CAPS Mode is OFF, enter the *string* as it appears in the text and the *replacement-string* as it should appear in the text. If CAPS Mode is ON, all lowercase characters are translated to uppercase characters.

## COPY Command

**COPY** [**S.**]clist-name[**OF** userid]

where:

**[S.]**-forces access to a global CLIST (if there is a CLIST of the same name under your user ID).

**clist-name**-the name of the CLIST from which you want to copy sourcelines.

**[OF userid]**-allows you to specify a CLIST belonging to another CA IDMS DMLO user ID.

Use the COPY command to copy sourcelines from another CLIST. To specify the location where lines will be copied, enter an 'A' (after) or 'B' (before) line command.

This command is available in Edit Mode.

## CURSOR Command

### CURSOR

The CURSOR command moves the cursor directly to the COMMAND line. It functions in the same way as the home key.

This command is available in Edit and Browse Modes.

The default keys are PF12 and PF24.

## DOWN Command

```
      [ / number-of-lines \ ]  
DOWN | < Max                > |  
      | | Half                | |  
      | \ Page                / |
```

where:

**Max**-specifies the last full screen at the bottom of the text.

**Half**-specifies to scroll down half a screen.

**Page**-specifies to scroll down a full screen.

The DOWN (scroll down) command is used to display source lines below your current view. The amount you scroll is determined by the Scroll setting. The setting can be overridden at any time.

This command is available in Edit and Browse Modes.

The default keys are PF8 and PF20.

## EDITOR-ID Command

### EDITOR-ID

The EDITOR-ID command displays the release number for the version of the Editor invoked. The release is displayed in message format.

This command is available in Edit and Browse Modes.

## ECHO Command

Use the ECHO command to preserve the primary command line. If ECHO is turned on, the last command entered on the command line is preserved and redisplayed. If ECHO is turned off, the last command entered is not preserved. The ECHO setting is maintained in the Editor profile for the signed on CA IDMS/DC user id. The PROFILE command can be used to display all environmental settings, which will include the ECHO setting. The syntax for the ECHO command is the following:

```
ECHO {ON  
      OFF}
```

**Default:** OFF

This command is available in edit and browse modes.

## END Command

**END**

Use the END command to save the current CLIST module (if changed) and return to the CA IDMS DMLO session mainline.

This command is available in Edit Mode.

## ENTER Command

**ENTER**

The ENTER command redisplay the current screen with any changes made.

This command is available in Edit and Browse Modes. The default key is ENTER.

## EXCLUDE Command

```
      r/ first-line \r  
EXCLUDE |< last-line >|  
      L\ ALL      /L
```

where:

**first line**-specifies that the first line number is to be excluded from the display.

**last-line**-specifies that the last line in the block of lines is to be excluded from the display. If this is left blank, the default is the last line in the text.

**ALL**-specifies that all lines in the text are excluded from the display.

The EXCLUDE command limits your display to specific lines within the text being edited. Redisplay excluded lines with the RESET command.

This command is available in Edit Mode.

## FIND Command

```
      r/      \r r/      \r r/      \r  
FIND |< ALL >| |< string >| | [Lb[rB]] |< X >|  
      L\ FIRST /L L\ *      /L L\ NX      /L
```

where:

**ALL**-specifies that all occurrences of a string are to be found in scanned lines.

**FIRST**-specifies that the first occurrence of a string is to be found.

**string**-specifies the string is to be found.

**\* (asterisk)**-specifies the string value from the last FIND command entered.

**lb rb**-specifies the left and right bounds (column positions) for the find. If specifying just the left bound, the string to found must begin in that same column. If specifying both left and right bounds, the string to be found can appear anywhere within those bounds.

**X**-specifies only excluded lines are to be scanned.

**NX**-specifies only non-excluded lines are to be scanned.

Use the FIND command to search for a string in the module.

The Editor begins searching at the position of the cursor when you enter the command. It searches downward until the string is found. If the cursor is on the COMMAND line when you enter the command, the Editor begins searching at the top line displayed.

The operands of this command can appear in any order.

This command is available in Edit Mode.

## FIRST Command

### **FIRst**

The FIRST command displays the first screen of the module.

This command is available in Edit and Browse Modes.

## KEYS Command

### **KEYS**

Use the KEYS command to view or change editor PF key settings.

This command is available in Edit and Browse Modes.

## LAST Command

### **LASt**

The LAST command presents the last screen of the module.

This command is available in Edit and Browse Modes.

## LEFT Command

**LEft** [**number-of-coLumns**]

The LEFT command scrolls the current display to the left the specified number of columns. If the number parameter is blank, the Scroll Options are used.

This command is available in Edit and Browse Modes.

## LOCATE Command

**Locate** **line-number**

where:

**line-number**-specifies the number of the line to which you want to move. The line you specify will be the top line displayed on the screen.

Use the LOCATE command to move the display to a specific source line or to the beginning or the end of the module.

This command is available in Edit and Browse Modes.

## Using the LOCATE Command

To move to a specific line, you specify the line number of the line you want displayed.

To move to the beginning of the module, you can specify 0 as the line number, and the first line of the module will be the top line displayed.

To move to the end of the module, you can specify the last line number or any larger number, and the last line of the module will be the top line displayed. For example, if the last line of the module is numbered 307 and you use 999, line number 307 will be the top line displayed.



## MEMORY Command

```
MEMory  r/          \r
         |< STATIC >|
         L\ DYNAMIC /L
```

where:

**STATIC**-specifies to obtain storage one time and track until the end of the session.

**DYNAMIC**-specifies to obtain new storage and free it each time the Editor driver module is called.

Internal storage is determined by the MEMORY command.

This command is available in Edit and Browse Modes.

## NULLS Command

```
NuLLs  r/          \r
         |< ON >|
         L\ OFF /L
```

The NULLS command is used to turn the NULLS Mode on and off. **ON** is the default. In the NULLS ON Mode, null characters replace all but the first blank in a line. If the line is completely blank, null characters are not substituted.

To easily use the keyboard insert mode key to insert characters, turn NULLS Mode ON so that edit will automatically insert trailing nulls in each data line in the display.

Normally, each data line is one field on the display. However, by using the TABS Mode, 3270 tab characters can be created in selected columns and is a way to break up a line into several fields. NULLS will replace trailing blanks in each field.

If edit places the cursor into a field, only blanks that follow the cursor on the line will be changed to NULLS. If a character is deleted with the DELETE key, all of the characters in the field are shifted left one position and a NULL character is inserted into the last position in the field. If the ERASE EOF key is pressed, NULL characters fill the field on which the cursor is located from the cursor to the end of the field.

This command is available in Edit Mode.

## PROFILE Command

### PROFIle

Use the PROFILE command to display the environmental parameters under which your Edit session is operating.

The PROFILE identifiers that are displayed correspond to the primary commands. When you change parameters that are unique to your profile, the changes are saved to the CA IDMS/DC user signon.

Use the RESET command to clear the display of any line commands, column markers, or extraneous messages.

This command is available in Edit and Browse Modes.

## RCHANGE Command

### RCHANGE

The RCHANGE command repeats the last CHANGE command that was executed.

The Editor begins searching at the first line of the display. When it reaches the bottom of data, the message "BOTTOM OF DATA REACHED" appears in the top line of the screen.

The default keys are PF6 and PF18.

This command is available in Edit Mode.

## Using the RCHANGE and RFIND PF Keys to Selectively

### Change Strings

You can use the RFIND PF key in conjunction with the RCHANGE PF key to selectively change strings. For example, consider the following sequences:

#### First Sample Sequence

Step	Command	Action
Part 1	CHANGE Work-Name-1 Work-Name-2	In Part 1, you enter the CHANGE command to change the next occurrence of Work-Name-1 to Work-Name-2.

Step	Command	Action
Part 2	RFIND key	In Part 2, you want to find the next occurrence of Work-Name-1, but you are not sure if you will want to change the string. By pressing the RFIND key, the next occurrence of Work-Name-1 which was specified in the CHANGE command during Part 1, will be found.
Part 3	RCHANGE key	In Part 3, you want to change the occurrence of Work-Name-1 that was found during Part 2 to Work-Name-2. By pressing the RCHANGE key, the occurrence will be changed.
Part 4	RFIND key RFIND key	In Part 4, you press the RFIND key to find the next occurrence of Work-Name-1. This time you do not want to change the string, so instead of pressing the RCHANGE key, you press the RFIND key again. The next occurrence of Work-Name-1 will be found.

#### Second Sample Sequence

Step	Command	Action
Part 1	CHANGE Work-Name-1 Work-Name-2 RFIND key	In Part 1, you want to find the next occurrence of Work-Name-1, but if you are not sure if you want to change it to Work-Name-2. If you key in the CHANGE command and press the RFIND key instead of the ENTER key, the RFIND will be executed. The next occurrence of Work-Name-1 that was specified in the CHANGE command will be found.
Part 2	RFIND key	In Part 2, you decide that you do not want to change the string that was found during Part 1, press the RFIND key. The next occurrence of Work-Name-1 will be found.
Part 3	RCHANGE key	In Part 3, you want to change the string that was found during Part 2, press the RCHANGE key. This changes Work-Name-1 to Work-Name-2.

## RESET Command

### RESet

Use the RESET command to clear the display of any line commands, column markers, or extraneous messages.

This command is available in Edit and Browse Modes.

The default keys are PF9 and PF21.

## RESHOW Command

### RESHOW

Use the RESHOW command to redisplay the original contents of a screen. This command is only valid when you have typed a screen of data but have *not* pressed the ENTER key.

**Note:** If you use the RESHOW Command you will overlay the current screen with the previous screen.

This command is available in Edit and Browse Modes.

## RFIND Command

### RFIND

The RFIND command repeats the last FIND command that was entered.

The RFIND search begins at the position of the cursor. When it reaches the end of the file, it will reach the bottom and the message line will state, BOTTOM OF DATA REACHED. Entering RFIND again will resume the search at the top of the file. Then if the string is not found in the file, the message line will state: **NO CHAR: *string* Found.** Entering the RFIND command has no effect.

To selectively change strings, use the RFIND PF key in conjunction with the **RCHANGE PF** key.

This command is available in Edit and Browse Modes.

The default keys are PF5 and PF17.

## RIGHT Command

**Right** [number-of-columns]

The RIGHT (scroll right) command scrolls the current display to the right the specified number of columns. If the number parameter is blank, the scroll options are used.

This command is available in Edit and Browse Modes.

The default keys are PF11 and PF23.

## SAVE Command

**SAVE**

Use the SAVE command to save the current CLIST module (if changed). The CLIST edit session will remain active.

This command is available in Edit Mode.

## TABB Command

**TABB**

The TABB (tab backward) command is used to move to the previous tab setting when TABS Mode is ON.

This command is invoked more efficiently if you assign it a PF key value.

This command is available in Edit Mode.

## TABF Command

**TABF**

The TABF (tab forward) command is used to move to the next tab setting when TABS Mode is ON.

This command is invoked more if you assign it a PF key value.

This command is available in Edit Mode.

## TABS Command

```
TABS  r/ ON          \r
      |< OFF        >|
      || tab-character ||
      \ \ operand  / \
```

where:

**tab-character**-specifies any character used to signify a tab.

**operand**-specifies any of the following and their settings:

- **ADS** specifies every five positions from 1 through 65.
- **ASM** specifies the positions 1, 10, 16, and 36.
- **COBOL** specifies the positions 8, 12, 16, and 20.
- **STND** specifies the positions 1, 10, 16, and 36.

The TABS command sets software tabbing. The commands TABF (ta b forward) and TABB (tab backward) are used to move a tab setting within the text. To customize tab settings, use the TABS line command.

This command is available in Edit Mode.

## TIME Command

### TIME

The TIME command displays the time-of-day and the date in the message area of the screen. The TIME is given in military hh:mm:ss format. The date is given in standard mm/dd/yy format.

This command is available in Edit and Browse Modes.

## TOP Command

### TOP

The TOP command displays the first full screen at the top of the source.

This command is available in Edit and Browse Modes.

## UP Command

```

      r/ number-of-Lines \r
UP |< Max                >|
   || Half              ||
   L\ Page              /J

```

where:

**number-of lines**-specifies the number of lines to scroll. If this is blank, then scrolling is determined by the Scroll Option.

**Max**-specifies that you want to scroll to the first screen of text.

**Page**-specifies that you want to scroll a full screen of text.

The UP (scroll up) command is used to display source lines above your current view. The amount you scroll is determined by the scroll option setting. The setting can be overridden at any time.

This command is available in Edit and Browse Modes.

The default keys are PF7 and PF19.

## Line Commands

Line commands are entered in Edit Mode with the cursor positioned to the left of the source lines, in the line number fields. To use a line command, type over the line numbers.

### Entering Line Commands

Line commands are entered within the line number at the left of the line data. A line command is considered to be any characters entered at or to the left of the cursor in the line sequence number fields.

## How to Use Line Commands

If you wanted to repeat the line 10 times, here is how the line would appear:

000003 Before entering R (repeat) Command

R10003 After entering R (repeat) Command

For the Editor to read the command as R10:

- Type 'R10' in the line number field and press the ENTER key.
- Position the cursor immediately after R10 (type 'R1' and move the cursor to the right one position) and press the ENTER key.

## A (after) Command

**A**

The A (after) line command is used in conjunction with the C (copy) and M (move) line commands, and the COPY primary command to copy another CLIST after the line containing the A (after) line command.

## B (before) Command

**B**

The B (before) line command is used in conjunction with the C (copy) and M (move) line commands and the COPY primary command to copy another CLIST before the line containing the B (before) line command.

## BNDS (bounds) Command

**BNDS**

The BNDS command displays and allows changes to the current boundary settings. The bounds line is displayed at the line where you entered the command.

Change the current bounds setting by using the < character to define the left bound and the > character to define the right bound.

To remove the bounds line from the display, use the D (delete) line command or the RESET primary command.



## COLS (columns) Command

### COLS

The COLS command displays a line with the column markings for you to use as a reference. This line is for reference purposes only. It is not given a line number and is not saved with the text.

The column markings line appears before the line in which you enter the COLS command.

To remove the COLS line from the display, use the D (delete) line command or the RESET primary command.

## C (copy) Command

### C[*number-of-lines*]

where:

**number-of-lines**-specifies the number of lines to be copied. The default is 1.

**C**-specifies a single line to be copied.

**Cn** - specifies the first of *n* lines to be copied.

**CC...CC**-specifies the first and last lines of a block of lines to be copied.

Use the C (copy) line command to copy one line or block of lines. The B (before) and A (after) line commands are used to specify the destination of the line or block to be copied. No other line commands can be entered on the lines to be copied.

## Rules for Using the Copy Line Command

- When using the C *number-of-lines* or the CC form of the command, you cannot enter any other commands on the lines being copied.
- Each CC must be paired with another CC.
- You must pair a B (before) or an A (after) line command with every C or pair of CC commands.

## D (delete) Command

**D[number-of-Lines]**

where:

**number-of-lines** specifies the number of lines to be deleted.

**D** specifies a single line to be deleted.

**Dn** specifies the first of *n* lines to be deleted.

**DD...DD** specifies the first and last lines of a block of lines to be deleted.

The D (delete) command is used to delete a line or block of lines. No other line commands can be entered on the lines to be deleted.

### Rules for Using the D (delete) Line Command

- When using the D *number-of-lines* or the DD form of the command, you cannot enter any other commands on the lines being deleted.
- Each DD must be paired with another DD.

## X (exclude) Command

**X[number-of-Lines]**

where:

**number-of-lines** specifies the number of lines excluded.

**X** specifies a single line to be excluded.

**Xn** specifies the first of *n* lines to be excluded.

**XX...XX** specifies the first and last lines of a block of lines to be excluded.

Use the X (exclude) command to exclude lines from the display.

## I (insert) Command

**I[*number-of-lines*]**

where:

**number-of-lines**-specifies the number of lines to be inserted. The default is 1.

Use the I (insert) command to insert blank lines after the line in which the I command is entered. The I command is not used with the A (after) and B (before) line commands. If no data is typed on an inserted line, the blank inserted line is deleted from the display after the ENTER key is pressed or the RESET, UP, or DOWN primary command is entered.

## M (move) Command

**M[*number-of-lines*]**

where:

**number-of-lines**-specifies the number of lines to be moved. The default is 1.

**M**-specifies a single line to be moved.

**Mn**-specifies the first of *n* lines to be moved.

**MM...MM**-specifies the first and last lines of a block.

Use the M (move) command to move a line or block of lines. The B (before) or A (after) line commands are used to specify the destination of the lines to be moved. No other commands can be entered on the lines to be moved.

### Rules for Using the Move Line Command

- When using the M *number-of-lines* or the MM form of the command, you cannot enter any other commands on the lines being moved.
- Each MM must be paired with another MM.
- You must pair a B (before) or an A (after) line command with every M or pair of MM commands.

## R (repeat) Command

**R[*number-of-times*]**

where:

**number-of-times**-specifies the number of times a line or block of lines is repeated. The default is 1.

**R**-specifies a single line to be repeated.

**Rn**-specifies a single line to be repeated *n* times.

**RR** -specifies the first and last lines of a block to be repeated one time.

The R (repeat) line command is used to repeat a line or block of lines directly after the last line to be repeated.

### Rules for Using the R (repeat) Line Command

- Pair each RR *number-of-lines* with another RR *number-of-lines* to complete a block command.
- If *number-of-lines* on the RR block commands differ, the greater number is used.
- No other line commands can be used on lines being repeated.

## TABS Command

Type TABS in the line number field to view the current tab settings. You can also use the TABS command to change the tabs by over striking the current setting (indicate by tab character with the new positions you choose).

The TABS line may be deleted from the display by the D (delete) line command or the RESET primary command.

## Text Manipulation Line Commands

The text manipulation line commands are used when entering or altering text data. These commands are especially useful when used in combinations. For example, use TS (text split), enter a word or phrase, and then use TF (text flow) to reformat the paragraph.

# Appendix B: Batch CLIST Processing

---

This section contains the following topics:

[Overview](#) (see page 205)

[Work Files](#) (see page 205)

[Batch CLIST Request Cards](#) (see page 207)

[Batch CLIST Upload File](#) (see page 208)

## Overview

This appendix provides a description of the batch CLIST processor, USDBCLST. This utility provides an easy way to migrate CLISTs between CA IDMS systems.

The batch CLIST processor supports the following functionality:

- Uploading one or more new CLISTs to the system.
- Printing one or more existing CLISTs from the system.
- Downloading one or more existing CLISTs from the system in a format which can be submitted directly for upload processing.
- Removing one or more existing CLISTs from the system.

The CA IDMS installation media contains the following sample JCL (intended for demonstration purposes only):

- USDBCLST – Contains a skeleton of the JCL required to invoke the batch CLIST processor.

## Work Files

The following datasets are used by the batch CLIST processor:

For z/OS

Input datasets

DD Name	Description	Usage	LRECL
SYSIPT	Request cards	Required for PRINT, DOWNLOAD, REMOVE	80

DD Name	Description	Usage	LRECL
UPLOAD	Source of CLISTs to add	Required for UPLOAD	80

## Output datasets

DD Name	Description	Usage	LRECL
REPORT	Source of printed CLISTs	Optional for all functions	133
AUDIT	Status of requests	Required for all functions	133
DOWNLOAD	Listings of downloaded CLISTs	Required for DOWNLOAD	80

## For z/VSE

## Input Datasets

DD Name	Logical Unit	Description	Usage	LRECL
SYSIPT	SYS015	Request Cards	Required for PRINT, DOWNLOAD, REMOVE	80
UPLOAD	SYS016	Source of CLISTs to add	Required for UPLOAD	80

## Output Datasets

DD Name	Logical Unit	Description	Usage	LRECL
REPORT	SYS017	Source of printed CLISTs	Optional for all functions	133
AUDIT	SYS018	Status of Requests	Required for all functions	133

DD Name	Logical Unit	Description	Usage	LRECL
DOWNLOAD	SYS019	Listings of downloaded CLISTs	Required for DOWNLOAD	80

## Batch CLIST Request Cards

Use request cards to download, print, or remove one or more CLISTs.

**Note:** Request cards are not required to upload a CLIST.

A request card describes the CLIST to be processed and the desired function. All variables must start in the appropriate columns. Values should be padded with spaces when necessary. The request card is described in detail below.

Columns	Variable	Description
01 - 08	<i>user-name</i>	Variable specifies a 1-8 character user name for whom the requested CLIST is stored
09 - 16	<i>clist-name</i>	Variable specifies a 1-8 character CLIST identifier
17 - 17	<i>operation code</i>	Variable specifies a 1 character operation code

The following table describes the operation code variable.

Code	Function	Description
P	Print only	Print the requested CLIST
D	Download only	Download the requested CLIST
X	Print and Download	Print and download the requested CLIST
R	Remove	Delete the requested CLIST

For example, the following request card will prompt the batch CLIST processor to delete the CLIST named SAMPLE belonging to user USER0001:

```
USER0001SAMPLE  R
```

The following request card prompts the batch CLIST processor to print the CLISTs SAMP1 and SAMP2 belonging to USER0002 in the report file. Additionally, the source for these CLISTs will be stored into the download file:

```
USER0002SAMP1  X  
USER0002SAMP2  X
```

## Batch CLIST Upload File

The upload file is used to add one or more CLISTs to the CA IDMS system. Each CLIST described in this file must begin with a parameter card which names the CLIST and assigns it to a user. Each request must end with an end card.

**Note:** No request cards are required to upload a CLIST. If a CLIST is already present on the system, the existing version is overwritten.

The parameter card is described below. All variables must start in the appropriate columns. Values should be padded with spaces when necessary.

Columns	Values	Description
01 - 01	'~' constant	Specifies a constant representing the start of a parameter or end card
02 - 09	user-name	Specifies a 1-8 character name of the user for whom the CLIST will be stored
10 - 17	clist-name	Specifies a 1-8 character CLIST identifier



One or more data cards follow a parameter card. Each data card contains one line of CLIST source. These lines should follow the standard CLIST conventions. Detailed information on CLIST command syntax can be found in [CA IDMS DMLO Session](#) (see page 21).

Every CLIST must end with an end card. The end card is described in detail below. All variables must start in the appropriate columns. Values should be padded with spaces when necessary.

Columns	Values
01 - 01	'~' constant
02 - 09	'**END**' constant

For example, the following lines coded in an upload file would prompt the batch CLIST processor to add CLISTs SAMP1 and SAMP2 belonging to USER0001:

```
~USER0001SAMP1
```

```
* OBTAIN CLIST
```

```
OBTAIN FIRST %1 WITHIN %2
```

```
ENDC
```

```
~**END**
```

```
~USER0001SAMP2
```

```
* STORE CLIST
```

```
STORE %1
```

```
ENDC
```

```
~**END**
```



# Index

---

## C

- CA IDMS DML Command Restrictions • 73
- CA IDMS DML Commands and Restrictions • 59
- CA IDMS DML Control Statements • 60
- CA IDMS DML Modification Statements • 65
- CA IDMS DML Online Session • 25
- CA IDMS DML Retrieval Statements • 61
- CA IDMS DML Save Statements • 66
- CA IDMS DML SCRATCH/QUEUE Processing • 69
- CA IDMS DMLO
  - description • 11
  - improving productivity • 12
  - using • 12
- CA IDMS DMLO Commands • 74
- CA IDMS DMLO Features • 13
- CA IDMS DMLO Session Mainline Screens • 32
- CA IDMS DMLO Session Termination • 46
- CA IDMS DMLO System Requirements • 119
- CLIST Commands • 52, 98
- CLIST Maintenance Commands • 95
- Command Processing Examples • 115

## E

- Editing Commands • 181
- Exit • 32, 122
- Extended Command Processing • 111

## I

- Installation • 121
- Introduction to the CA IDMS DMLO Session • 24

## L

- Line Commands • 182
- LRF Commands • 67

## M

- Messages Generated by CA IDMS DMLO • 125
- Mixed page group support • 19

## O

- Online Documentation Print Utility • 120

## P

- PF Key Processing • 106
- Primary Commands • 182, 185
- PROFILE Maintenance Commands • 91

## S

- SET DUMP • 86
- SET NODUMP • 86
- Syntax Notation • 58

## T

- Text Manipulation Line Commands • 204

## U

- User Exit Module • 20, 122
- Using the CA IDMS DMLO CLIST Processor • 47

## V

- Variable Symbols • 107