CA-IDMS®

Command Facility
15.0

Computer Associates™

This documentation and related computer software program (hereinafter referred to as the “Documentation”) is for the end user's informational purposes only and is subject to change or withdrawal by Computer Associates International, Inc. (“CA”) at any time.

This documentation may not be copied, transferred, reproduced, disclosed or duplicated, in whole or in part, without the prior written consent of CA. This documentation is proprietary information of CA and protected by the copyright laws of the United States and international treaties.

Notwithstanding the foregoing, licensed users may print a reasonable number of copies of this documentation for their own internal use, provided that all CA copyright notices and legends are affixed to each reproduced copy. Only authorized employees, consultants, or agents of the user who are bound by the confidentiality provisions of the license for the software are permitted to have access to such copies.

This right to print copies is limited to the period during which the license for the product remains in full force and effect. Should the license terminate for any reason, it shall be the user's responsibility to return to CA the reproduced copies or to certify to CA that same have been 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 the end user or any third party for any loss or damage, direct or indirect, from the use of this documentation, including without limitation, lost profits, business interruption, goodwill, or lost data, even if CA is expressly advised of such loss or damage.

The use of any product referenced in this documentation and this documentation is governed by the end user's applicable license agreement.

The manufacturer of this documentation is Computer Associates International, Inc.

Provided with “Restricted Rights” as set forth in 48 C.F.R. Section 12.212, 48 C.F.R. Sections 52.227-19(c)(1) and (2) or DFARS Section 252.227-7013(c)(1)(ii) or applicable successor provisions.

Second Edition, October 2001

© 2001 Computer Associates International, Inc.
All rights reserved.

All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

Contents

How to Use This Manual v

Chapter 1. Overview of the CA-IDMS Command Facility 1-1
1.1 What is the command facility? 1-4
1.1.1 Types of statements you can submit 1-4
1.2 Coding considerations 1-6
1.3 Special coding for command facility statements 1-7
1.4 Connecting to a dictionary 1-8
1.5 Using SET OPTIONS to select options 1-9

Chapter 2. Batch Processing 2-1
2.1 Overview of IDMSBCF 2-4
2.1.1 Description of processing 2-4
2.1.2 IDMSBCF job stream 2-4
2.1.3 Batch output 2-4
2.1.4 Return codes 2-4
2.2 JCL for OS/390 2-5
2.3 Commands for VM/ESA 2-8
2.4 JCL for VSE/ESA 2-10
2.5 JCL for BS2000/OSD 2-17

Chapter 3. Online Processing 3-1
3.1 Overview of OCF 3-4
3.2 Beginning an OCF session 3-5
3.2.1 OCF screen format 3-5
3.3 Conducting an online session 3-7
3.4 Ending an OCF session 3-8
3.5 Executing statements dynamically 3-9
3.6 Executing statements stored in modules (OCFX) 3-10
3.6.1 Executing at the system prompt 3-10
3.7 Executing as part of a command list 3-12
3.7.1 OCFX status codes 3-12
3.8 Creating, editing, and saving modules 3-14
3.9 Commands for session control 3-15
3.9.1 Summary of session-control commands 3-15
3.10 Function keys assigned to session-control commands 3-17
3.10.1 APPLY 3-18
3.10.2 CLEAR 3-18
3.10.3 DISPLAY LINE 3-18
3.10.4 DISPLAY PAGE 3-19
3.10.5 END 3-20
3.10.6 ENTER= 3-20
3.10.7 ESCAPE 3-21
3.10.8 FIND 3-21
3.10.9 HELP 3-21
3.10.10 INSERT 3-22
3.10.11 PRINT 3-22

3.10.12 REPEAT 3-23
3.10.13 RESHOW 3-23
3.10.14 SUSPEND 3-23
3.10.15 SWAP 3-24
3.10.16 SWITCH 3-24
3.10.17 UPDATE 3-25
3.11 Commands for editing 3-26

Appendix A. Using the Command Facility as a Subprogram A-1
A.1 Overview A-3
A.2 Compiler interface parameter list A-5
A.2.1 CIO block A-5
A.2.2 CIOF block A-6
A.2.3 User parameters A-7
A.3 Work-area file A-8
A.4 Sample program that calls the command facility A-9

Index X-1

How to Use This Manual

What this manual is about

This document provides information on the two components of the CA-IDMS Command Facility:

- The batch command facility — allows you to submit command statements as part of a batch job stream
- The online command facility — allows you to submit command statements interactively and see the resulting output on a display screen

The purpose of the document is to describe these two components and how to use them to submit these types of CA-IDMS statements:

- Physical DDL statements
- Logical DDL statements (SQL-defined databases only)
- SQL DML statements
- Utility statements
- Security administration statements
- Command facility statements (as an alternative to the command facility compiler)

Who should use this document

Users of CA-IDMS products. In particular:

- Database administrators
- Data dictionary administrators
- Programmers
- Those responsible for defining physical and logical databases
- Those responsible for the security administration of a CA-IDMS/DB database
- Those using SQL data manipulation language (DML) statements to access and/or alter a CA-IDMS/DB database

How information is presented

This document has three chapters:

- Chapter 1 — Overview of the CA-IDMS Command Facility

  Describes the CA-IDMS Command Facility and its online and batch components; provides information on coding guidelines and describes the SET OPTIONS statement you can use to control processing and output; also describes status messages for the command facility

- Chapter 2 — Batch Processing

  Describes how to use an IDMSBCF job stream to submit statements in a batch environment

- Chapter 3 — Online Processing

  Describes how to submit and execute statements in an online OCF environment; also describes how to execute dictionary modules at the system prompt or as part of a system command list (CLIST)

Related documentation

Use this document in conjunction with these documents:

- *CA-IDMS Database Administration*
- *CA-IDMS Utilities*
- *CA-IDMS Security Administration*
- *CA-IDMS SQL Reference Guide*
- *CA-IDMS IDD DDDL Reference Guide*

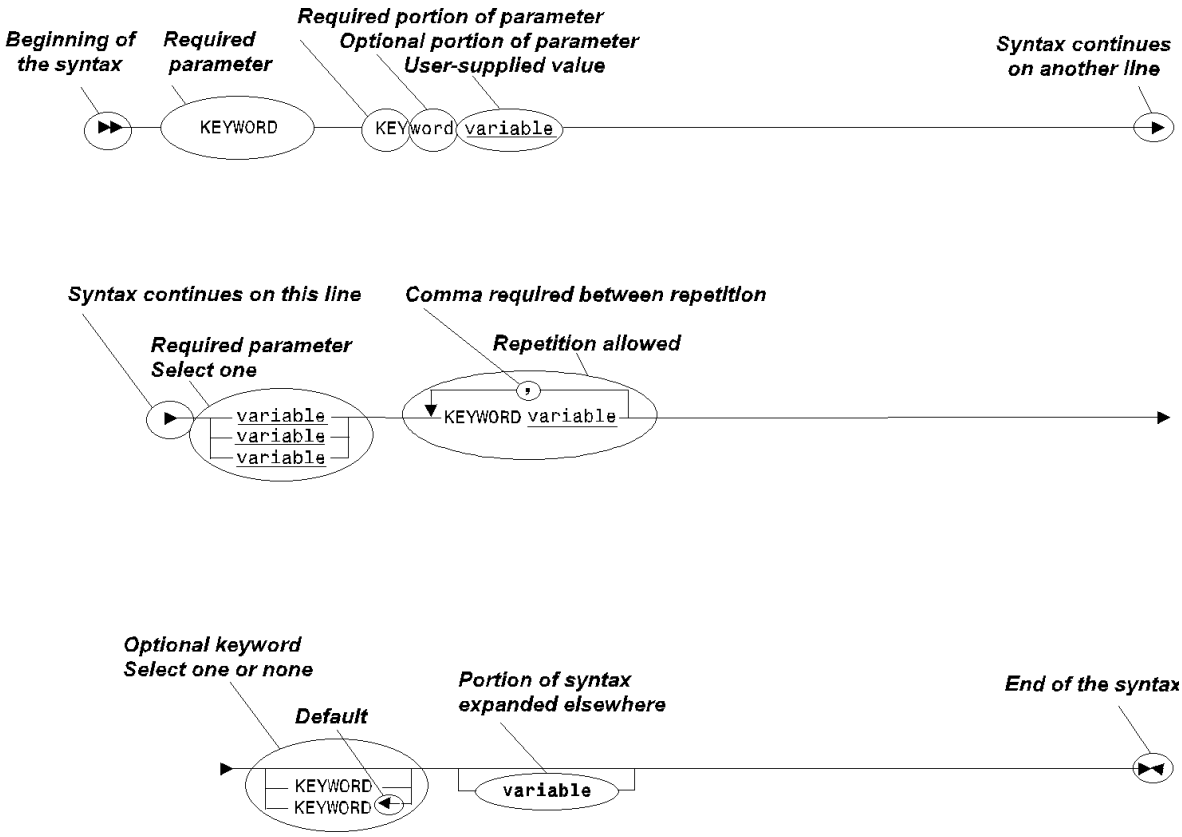
Understanding syntax diagrams

Look at the list of notation conventions below to see how syntax is presented in this manual. The example following the list shows how the conventions are used.

`UPPERCASE`
`OR`
`SPECIAL CHARACTERS`
	Represents a required keyword, partial keyword, character, or symbol that must be entered completely as shown.

`lowercase`	Represents an optional keyword or partial keyword that, if used, must be entered completely as shown.
`underlined lowercase`	Represents a value that you supply.
←	Points to the default in a list of choices.
`lowercase bold`	Represents a portion of the syntax shown in greater detail at the end of the syntax or elsewhere in the document.
►►	Shows the beginning of a complete piece of syntax.
►◄	Shows the end of a complete piece of syntax.
►	Shows that the syntax continues on the next line.
►	Shows that the syntax continues on this line.
►	Shows that the parameter continues on the next line.
►	Shows that a parameter continues on this line.
►— parameter —►	Shows a required parameter.
►┬ parameter ┬► └ parameter ┘	Shows a choice of required parameters. You must select one.
►┬ ┬► └ parameter ┘	Shows an optional parameter.
►┬ ┬► ├ parameter ┤ └ parameter ┘	Shows a choice of optional parameters. Select one or none.
┌ ┐ ►─▼─ parameter ─┴─►	Shows that you can repeat the parameter or specify more than one parameter.
┌ , ┐ ►─▼─ parameter ─┴─►	Shows that you must enter a comma between repetitions of the parameter.

Sample syntax diagram



Chapter 1. Overview of the CA-IDMS Command Facility

1.1 What is the command facility? 1-4
1.1.1 Types of statements you can submit 1-4
1.2 Coding considerations 1-6
1.3 Special coding for command facility statements 1-7
1.4 Connecting to a dictionary 1-8
1.5 Using SET OPTIONS to select options 1-9

This chapter covers the following topics:

- A description of the command facility
- A description of the types of statements you can submit using the command facility
- Guidelines for coding statements that you submit
- Information on selecting processing and formatting options for the command facility session (using the SET OPTIONS statement)
- Command facility status messages

1.1 What is the command facility?

The command facility is a CA-IDMS tool that you use to submit several types of CA-IDMS statements.

Batch and online components: The command facility is available for submitting statements using a batch or online component:

- **The batch command facility (IDMSBCF)** allows you to submit command statements as part of a batch job stream
- **The online command facility (OCF)** allows you to submit command statements interactively and see the resulting output on a display screen

1.1.1 Types of statements you can submit

This section describes the types of statements you can submit using the CA-IDMS Command Facility.

Physical DDL statements: Use Physical DDL statements for SQL and non-SQL defined databases to define:

DMCLs

Segments

Database name tables

Logical DDL statements: Use logical DDL statements for SQL-defined databases to define:

Schemas

Tables

Indexes

CALC keys

Referential constraints

Views

SQL DML statements: Use SQL DML statements to access SQL-defined and non-SQL defined databases and:

Update data

Insert rows

Delete rows

Select rows

Utility statements: Use the batch command facility to enter utility statements.

Security administration statements: Use security administration statements to secure:

User and group access

User profiles

Database entities

System entities

Command facility statements: As an alternative to using the command facility compiler to submit command facility statements, you can use the command facility to submit these statements.

►► See 1.3, “Special coding for command facility statements” on page 1-7 for information about using the command facility to submit command facility statements.

1.2 Coding considerations

Follow the coding guidelines described in the following table when you use the CA-IDMS Command Facility.

Guideline for:	Description
Input	Must be in the columns ranging from 1 to 71 (batch), columns ranging from 1-79 (online). Can be either uppercase or lowercase characters (or mixed). A statement can span any number of input lines. You can submit multiple statements on the same input line.
Syntax	Use the syntax rules that apply to the type of statement: ■ Physical DDL — See *CA-IDMS Database Administration* ■ Logical DDL (SQL-defined databases) — See *CA-IDMS SQL Reference Guide* ■ SQL DML — See *CA-IDMS SQL Reference Guide* ■ Utilities — See *CA-IDMS Utilities* ■ Security administration — See *CA-IDMS Security Administration*
Statement terminator	End each statement with a semicolon. The semicolon can be on the line with the statement, or it can be on the line following the statement. You can omit the terminator at the end of your last statement.
Comments	Begin comments with a double hyphen (--). Comments continue to the end of an input line.
Connecting to a dictionary	Use a CONNECT statement in a batch or online command facility session (see 1.4, “Connecting to a dictionary” on page 1-8)
Session control	Use the SET OPTIONS statement (see 1.5, “Using SET OPTIONS to select options” on page 1-9)
Output formatting control	Use the SET OPTIONS statement (see 1.5, “Using SET OPTIONS to select options” on page 1-9)

1.3 Special coding for command facility statements

As an alternative to switching to the command facility compiler, you can submit command facility statements using the command facility. You can submit command facility statements either singly or as a block of statements. These two methods are described below, as well as considerations for when you need to submit a command facility SIGNON statement.

Submitting a single command facility statement: Submit the statement as you would to the command facility compiler, except end each statement with a semicolon. Each statement is processed in a separate call to the command facility compiler.

If an entity type is ambiguous, precede it in syntax with the keywords command facility. For example, to differentiate a COMMAND FACILITY FILE from a DDL FILE, add the keywords command facility:

```
add command facility file name is ...
display all command facility files.
```

You can mix command facility statements with other statements you submit using the command facility.

Submitting a block of command facility statements: If you want to enter several command facility statements without making several calls to the command facility compiler, you can submit the command facility statements as a block. Place the command EXECIDD on a separate line preceding the command facility statements, and place the command ENDIDD on a separate line following the command facility statements. You can use a period or semicolon to terminate each command facility statement.

The following is an example of submitting multiple command facility statements for processing in a single call to the command facility compiler:

```
EXECIDD
add attribute sql within class language.
add module sqlsel language sql
   module source follows
       select * from employee
       where proj_leader_id = emp_id
       and proj_id = project_number;
   msend.
ENDIDD
```

When to enter a COMMAND FACILITY SIGNON statement: Appropriate COMMAND FACILITY SIGNONs are generated when you enter command facility statements singly or as a block.

However, when your signon user ID for a command facility secured dictionary does not match your signon user ID for the system, enter a COMMAND FACILITY SIGNON statement as the first statement following EXECIDD. This signon must include the user ID and password required to access the secured dictionary.

1.4 Connecting to a dictionary

Specifying a dictionary: You can explicitly connect to a dictionary using the CONNECT statement.

Note: CONNECT is the OCF counterpart of a SIGNON statement, like COMMAND FACILITY SIGNON.

```
►►── CONNECT TO dictionary-name ──────────────────────────────────►◄
```

Dictionary-name is the name of the dictionary to be accessed during the batch or online command facility session; specifies the dictionary where definitions are stored.

Connecting automatically: If you do not use the CONNECT statement, the command facility automatically connects to the current dictionary, as follows:

- **Batch** — current dictionary is determined by the DICTNAME parameter in the SYSIDMS parameters file

  ►►See the *CA-IDMS Database Administration*, for more information on SYSIDMS parameters.

- **Online** — current dictionary is determined according to how it is set by the user:
  - DCUF SET DICTNAME
  - User signon profile
  - COMMAND FACILITY SIGNON

  ►► For information about DCUFs, refer to *CA-IDMS System Tasks and Operator Commands*. For information about the user signon profile, refer to *CA-IDMS Security Administration*. For information about COMMAND FACILITY SIGNON, refer to *CA-IDMS IDD DDDL Reference Guide*.

1.5 Using SET OPTIONS to select options

You can optionally include a SET OPTIONS statement in an IDMSBCF (batch) or OCF (online) session to:

- Control the processing of statements
- Format the output from the execution of SQL SELECT statements

Syntax: SET OPTIONS statement

```
                  ┌─────────────────────────┐
►►─── SET OPTIONS ─↓─┬─ session-option ─┬─┴─ ; ──────────────────────────►◄
                     └─ format-option ──┘
```

Expansion of session-option

```
►►─┬─────────────────────────────────────────────────────┬────────────────►◄
   └─┬─ AUTOCOMMIT ─┬─ ON ← ──┬──────────────┬─┬─┬─┘
     │              │         └─ CONTINUE ─┘ │ │
     │              ├─ COMMAND ─────────────┤ │
     │              └─ OFF ─────────────────┘ │
     └─ ON ERROR ─┬─ END ───────────┬─────────┘
                  ├─ CONTINUE ← ─┤
                  └─ ROLLBACK ───┘
```

Expansion of format-option

```
►►─┬───────────────────────────────────────────────────────────────────┬──►◄
   └─┬─ TITLE ─┬─ 'title' ─┬─────────────────────────────────────────┤
     │         └─ DEFAULT ─┘                                         │
     ├─ HEADINGS ─┬─ OFF ─┬──────────────────────────────────────────┤
     │            └─ ON ← ┘                                          │
     ├─ LINES ─┬─ 60 ← ──────┬───────────────────────────────────────┤
     │         └─ line-count ┘                                       │
     ├─ WIDTH ─┬─ PAGE character-count ──────────────────────────────┤
     │         ├─ CHARacter ─┬─ 80 ← ──────────────────────┬───────┤
     │         │             └─ maximum-character-count ─┘       │
     │         ├─ NUMeric maximum-character-count ──────────────┤
     │         └─ COLUMN ─┬─ column-number  maximum-character-count ─┬─┘
     │                    └─ OFF ─────────────────────────────────┘
     ├─ UNDERLINE underline-character ───────────────────────────────┤
     ├─ PAGEBREAK ─┬─ column-number ─┬───────────────────────────────┬─┤
     │             │                 └─ column-number ─┬─────────────────┬─┘ │
     │             │                                   └─ column-number ─┘   │
     │             └─ OFF ───────────────────────────────────────────┤
     ├─ STATUS ─┬─ OFF ─┬────────────────────────────────────────────┤
     │          └─ ON ← ┘                                            │
     ├─ COMPRESS ─┬─ ON ──┬──────────────────────────────────────────┤
     │            └─ OFF ←┘                                          │
     └─ COLUMN WRAP ─┬─ OFF ─┬───────────────────────────────────────┘
                     └─ ON ← ┘
```

Parameters

session-option

On a SET OPTIONS statement, specifies session control options such as when to commit and rollback work and whether to continue processing when a statement execution results in an error.

Note: Session options don't apply to utility statements.

AUTOCOMMIT

Specifies AUTOCOMMIT options after successful statement execution (see the ON ERROR parameter for information about options for statements that return errors). AUTOCOMMIT options are shown in the following table.

Note: AUTOCOMMIT does not apply to utility statements.

Option	IDMSBCF	OCF	OCFX
AUTOCOMMIT ON (default)	Executes a COMMIT WORK RELEASE after the last statement in the input stream.	Executes a COMMIT WORK CONTINUE at the end of the edit buffer. Executes a COMMIT WORK RELEASE at the end of the SQL session; resources are released and temporary tables are dropped.	Executes a COMMIT WORK RELEASE at the end of the module.
AUTOCOMMIT ON CONTINUE	N/A	Holds resources until a COMMIT is issued, enabling you to keep temporary tables; note that this option can slow performance and should be used with caution.	N/A
AUTOCOMMIT COMMAND	Executes a COMMIT WORK CONTINUE after each successfully executed statement.		
AUTOCOMMIT OFF	No automatic COMMIT occurs; you must specify COMMIT WORK RELEASE manually. **IDMSBCF and OCFX** — A release is issued at the end of processing, and this may cause a rollback if any work has not been committed.		

ON ERROR

Specifies whether the session ends when a statement returns an error status code (a code with a value less than zero). Options are shown in the following table.

Note: ROLLBACK does not apply to utility statements.

Option	IDMSBCF	OCF	OCFX
ON ERROR CONTINUE (default)	Continues executing.	Continues executing.	Continues executing.
ON ERROR END	Ends job step and session. Proceeds with the AUTOCOMMIT option.	Ends execution of commands in the current edit buffer and ends the SQL session. Does not end the OCF session. Proceeds with the AUTOCOMMIT option.	Ends execution of commands in the module and ends the session. Proceeds with the AUTOCOMMIT option.
ON ERROR ROLLBACK	Issues a ROLLBACK RELEASE and ends job step.	Issues a ROLLBACK RELEASE and ends the SQL session. Does not end the OCF session.	Issues a ROLLBACK RELEASE and ends the execution of the module.

format-option

On a SET OPTIONS statement, specifies options for formatting the output of an SQL SELECT statement.

TITLE

Specifies a user-defined title line or resets the default title line. The TITLE parameter applies to IDMSBCF only.

'*title*'

Overrides the default title line with the specified title. You must enclose a title in single quotes.

DEFAULT

Resets the title line to the default. The default title line is 'IDMS Batch Command Facility'.

HEADINGS

Enables or suppresses page breaks and the output formatting that goes with them.

OFF
Suppresses page breaks. The title and column headings appear only once. Use this parameter when you intend to browse batch output online. OFF is the OCF default.

ON
Enables page breaks so that column name headings appear at the top of each page. ON is the IDMSBCF default.

LINES *line-count*
Sets the number of lines per page (the default is 60).

WIDTH
Sets page and/or column widths, or resets default widths.

PAGE *character-count*
Sets the width of a page for output. The range for IDMSBCF or OCFX is 40 to 132 (anything less than 40 is set to 40, anything greater than 132 is set to 132). The maximum value you should specify for OCF is 76; this reserves three characters for the *+ and space character preceding output; to suppress these characters, specify WIDTH PAGE 79.

Note: The PAGE parameter overrides the CHARACTER parameter.

CHARacter *maximum-character-count*
Sets the maximum width of *non-numeric* columns. Range is 1 to the width of the page (the default is 80). Columns longer than *maximum-character-count* are wrapped within the line. WIDTH CHAR must be less than the value specified for WIDTH PAGE.

NUMeric *maximum-character-count*
Sets the maximum width of *numeric* columns. Range is 1 to 32 (the default is the maximum length for the data type). If an exact value (including the sign) contains more digits than *maximum-character-count*, the value is replaced in output by a string of asterisks. Approximated values are rounded to fit the specified width of the column.

COLUMN *column-number maximum-character-count*
Sets the maximum width of the column identified by *column-number*. The maximum value you can specify for *maximum-character-count* is the width of the page. This parameter overrides WIDTH CHARACTER and WIDTH NUMERIC parameters for the specified column.

COLUMN OFF
Turns off all column width settings that were specified using the WIDTH COLUMN parameter.

UNDERLINE *underline-character*
Specifies the character that is to be used to underline column headings in output. The default is a blank space.

PAGEBREAK
Enables or disables settings required for a page break.

column-number
Specifies the number of an output column that is to be used to control page breaks. A change in the value of a column forces a page break. You can specify up to three column numbers. The HEADING OFF and COLUMN WRAP ON parameters disable this parameter.

OFF
Clears the column settings specified in the PAGEBREAK parameter of a previous SET OPTIONS statement.

STATUS
Controls the display of return codes.

OFF
Specifies that only non-zero status codes are to be displayed after statement execution.

ON
Specifies that return codes are to be displayed after the execution of every statement. ON is the default.

COMPRESS
Controls truncation of the output resulting from the execution of an SQL SELECT statement.

ON
Specifies truncation of output columns so that an entire row fits on one line.

OFF
Specifies normal output without any truncation of lines. OFF is the default.

COLUMN WRAP
Enables or disables line wrap for output columns.

OFF
Specifies that non-numeric columns that are longer than the available output space are to be truncated.

ON
Specifies that non-numeric columns that are longer than the available output space are to be displayed on multiple lines. ON is the default unless the COMPRESS ON parameter is used.

Usage:

Statement terminator: Use a semicolon to terminate a SET OPTIONS statement.

Number of SET OPTIONS statements in a job step: You can use more than one SET OPTIONS statement in an IDMSBCF or OCF session. This enables you to change session and formatting options without requiring you to end the session.

Each parameter you specify remains in effect to the end of the session unless you explicitly change that same parameter in a subsequent SET OPTIONS statement.

Page breaks for ordered information: You can use the PAGEBREAK parameter of SET OPTIONS to separate information sorted by the ORDER BY clause of the SQL SELECT statement.

Examples of SET OPTIONS

Session control parameters: The following SET OPTIONS statement specifies that a COMMIT WORK CONTINUE is to be issued after the successful execution of each statement; ON ERROR END specifies that the session is to end if the execution of a statement results in an error.

```
set options autocommit command
        on error end;
```

Submitting batch and viewing online: The following example shows SET OPTIONS parameters that you might typically use to submit an IDMSBCF batch job and view the output online.

```
set options title 'employee list'
        headings on
        underline *
        lines 24
        width page 80
        width char 10
        width num 5
        width column 2 6
        pagebreak 2 4
        column wrap off;
```

With these parameters specified, the output is formatted as follows:

- Headings appear at the top of each page and are underscored using asterisks (underline *)
- The page length is set to the number of lines on the screen (24)
- The page width is set to the width of the screen
- Non-numeric and numeric output columns are set to 10 and 5, respectively, with a maximum width in column 2 of 6 characters
- Any change in the value of column 2 or column 4 will force a new page
- Non-numeric columns will be truncated as needed so output rows can appear on a single line (WRAP OFF)

Effects of WIDTH PAGE and COMPRESS ON: The following example shows the output of a SELECT statement when the output exceeds the value specified by WIDTH PAGE. Because the screen isn't wide enough to display all four columns of data, the data for the fourth column is displayed after all of the data for the first three columns is displayed.

```
                    OCF 15.0 ONLINE IDMS PAGE 1 LINE 1          1/137

SELECT EMP_LNAME,EMP_FNAME,D.DEPT_ID, DEPT_NAME
FROM DEMOEMPL.DEPARTMENT D,DEMOEMPL.EMPLOYEE E
WHERE D.DEPT_ID=E.DEPT_ID;
*+
*+ EMP_LNAME             EMP_FNAME             DEPT_ID
*+ ---------             ---------             -------
*+ Brooks                John                     3510
*+ Park                  Deborah                  2210
*+ Smith                 Carl                     3530
*+ Spade                 Samuel                   4600
*+ Loren                 Martin                   4600
*+ Anderson              Alice                    6200
*+ MacGregor             Bruce                    2200
*+ Lynn                  David                    2200
*+
*+ DEPT_NAME
*+ ---------
*+ APPRAISAL - USED CARS
*+ SALES - NEW CARS
*+ APPRAISAL - SERVICE
*+ MAINTENANCE
*+ MAINTENANCE
*+ CORPORATE ADMINISTRATION
*+ SALES - USED CARS
*+ SALES - USED CARS
*+ 8 rows processed
```

As an alternative, you can specify COMPRESS ON. This truncates column output as needed so that all output can fit on a single line.

If you specify too many columns in the SELECT statement, COMPRESS ON may not work. In this case, an error message will alert you.

Effects of WIDTH CHARACTER and COLUMN WRAP OFF: The following example shows how output may be displayed when WIDTH CHARACTER is specified. The maximum width of non-numeric columns is set to 10. This causes the output for the 20-character JOB_TITLE column to wrap to subsequent output lines, as needed.

```
                        OCF 15.0 ONLINE IDMS NO ERRORS                1/59
SET OPTIONS WIDTH CHAR 10;
*+ Status = 0
SELECT
JOB_ID, JOB_TITLE, MIN_RATE, MAX_RATE, EFF_DATE FROM JOB;
*+
*+ JOB_ID  JOB_TITLE         MIN_RATE        MAX_RATE  EFF_DATE
*+ ------  ---------         --------        --------  --------
*+   4666  Sr Mechani        20500.00        45500.00  1979-06-01
*+         c
*+   5555  Salesperso        15000.00        39500.00  1975-01-01
*+         n
*+   4123  Recruiter         17500.00        28000.00  1985-03-01
*+
*+   4025  Writer - M        15500.00        25000.00  1987-06-01
*+         ktng
*+   4023  Accountant        22000.00        60000.00  1976-01-01
*+
*+   8001  Vice Presi        45000.00        68000.00  1979-01-01
*+         dent
*+   2077  Purch Cler         8500.00        15000.00  1980-03-01
*+         k
*+   2051  AP Clerk              4.80           10.60  1980-03-01
*+
 .
 .
 .
```

Additional space for wrapping the 20 characters in JOB_TITLE is provided even when the column output does not require it. If you specify COLUMN WRAP OFF, the non-numeric columns will be truncated instead of wrapping.

Chapter 2. Batch Processing

2.1 Overview of IDMSBCF 2-4
2.1.1 Description of processing 2-4
2.1.2 IDMSBCF job stream 2-4
2.1.3 Batch output 2-4
2.1.4 Return codes 2-4
2.2 JCL for OS/390 2-5
2.3 Commands for VM/ESA 2-8
2.4 JCL for VSE/ESA 2-10
2.5 JCL for BS2000/OSD 2-17

IDMSBCF is the batch processing component of the command facility. This chapter provides the following information on the IDMSBCF batch environment:

- A definition of IDMSBCF
- A description of IDMSBCF processing
- Information on IDMSBCF output
- JCL/commands for operating systems

2.1 Overview of IDMSBCF

IDMSBCF is the batch component of the command facility. To submit statements in a batch environment, you enter the statements in an IDMSBCF job stream.

2.1.1 Description of processing

Unless you specify otherwise in a SET OPTIONS statement, IDMSBCF processes statements one at a time, and a COMMIT WORK RELEASE is issued after the last statement in the input stream.

▸▸ For information on the SET OPTIONS statement, see 1.5, “Using SET OPTIONS to select options” on page 1-9.

2.1.2 IDMSBCF job stream

An IDMSBCF session consists of a single job step in a batch job stream. In this job step you include:

- A declaration of the program IDMSBCF
- All of the statements you want to execute, including optional SET OPTIONS statements

2.1.3 Batch output

Output is assigned to the file you specify in the IDMSBCF job stream. Output from the execution of a statement includes:

- A redisplay of the statement(s) you executed
- Status messages (error messages, warnings, system messages)

Output from the execution of an SQL SELECT statement might additionally include:

- Results of execution
- A message stating how many SQL table rows were processed

2.1.4 Return codes

The following return codes can be issued by the IDMSBCF command processing program:

- 00 - All commands were processed. No errors or warnings were issued.
- 04 - All commands were processed. One or more commands had warnings issued.
- 08 - All commands were processed. One or more commands had errors and were not completely executed.
- 12 - An internal error has occurred. Other messages should have been issued.

2.2 JCL for OS/390

Executing under the central version: Use the following JCL for an IDMSBCF session that will run under the central version.

IDMSBCF (OS/390)

```
//stepname EXEC PGM=IDMSBCF,REGION=1024K
//STEPLIB  DD   DSN=idms.dba.loadlib,DISP=SHR
           DD   DSN=idms.loadlib,DISP=SHR
//sysctl   DD   DSN=idms.sysctl,DISP=SHR
//dcmsg    DD   DSN=idms.sysmsg.ddldcmsg,DISP=SHR
//SYSLST   DD   SYSOUT=A
//SYSIDMS  DD   *
DMCL=dmcl-name
DICTNAME=dictionary-name
Other SYSIDMS parameters, as appropriate
/*
//SYSIPT   DD   *
Statements (including optional SET OPTIONS statements)
/*
```

▸▸See the *CA-IDMS Database Administration*, for more information on SYSIDMS parameters.

`stepname`	Name of the job step
`idms.dba.loadlib`	Dataset name the load library containing the DMCL and database name table load modules
`idms.loadlib`	Dataset name the load library containing the CA-IDMS executable modules
`sysctl`	Ddname of the SYSCTL file
`idms.sysctl`	Dataset name of the SYSCTL file (the link to the central version)
`dcmsg`	Ddname of the system message (DDLDCMSG) area
`idms.sysmsg.ddldcmsg`	Dataset name of the system message (DDLDCMSG) area
`dmcl-name`	Name of the DMCL
`dictionary-name`	Name of the dictionary to access You can specify DICTNAME as a SYSIDMS parameter or you can use the CONNECT TO *dictname* statement. CONNECT TO supersedes the SYSIDMS parameter.

If you submit a PUNCH statement: If you are going to submit any PUNCH statements, include a SYSPCH statement in JCL, for example:

```
//SYSPCH  DD DSN=dsname, DISP=(NEW,KEEP,DELETE),
             DCB=(RECFM=FB,BLKSIZE=9040,LRECL=80),
                  SPACE=space-specification,
                  UNIT=unit,VOL=SER=nnnnnn
```

If you submit physical DDL statements: If you are going to submit any physical DDL statements, include the following DD statements:

```
//dccat    DD   DSN=idms.system.ddlcat,DISP=SHR
//dccatl   DD   DSN=idms.system.ddlcatlod,DISP=SHR
//dccatx   DD   DSN=idms.system.ddlcatx,DISP=SHR
```

dccat	Ddname of the system dictionary catalog (DDLCAT) area
idms.system.ddlcat	Dataset name of the system dictionary catalog (DDLCAT) area
dccatl	Ddname of the system dictionary catalog load (DDLCATLOD) area
idms.system.ddlcatlod	Dataset name of the system dictionary catalog load (DDLCATLOD) area
dccatx	Ddname of the system dictionary catalog index (DDLCATX) area
idms.system.ddlcatx	Dataset name of the system dictionary catalog index (DDLCATX) area

If you submit SQL statements: If you are going to submit any SQL DDL or SQL DML statements, include the following DD statements:

```
//sqldd    DD DSN=idms.syssql.ddlcat,DISP=SHR
//sqllod   DD DSN=idms.syssql.ddlcatlod,DISP=SHR
//sqlxdd   DD DSN=idms.syssql.ddlcatx,DISP=SHR
//userdb   DD DSN=user.userdb,DISP=SHR
```

sqldd	Ddname of the SQL catalog area
idms.syssql.ddlcat	Dataset name of the SQL catalog area
sqllod	Ddname of the SQL catalog load area
idms.syssql.ddlcatlod	Dataset name of the SQL catalog load area
sqlxdd	Ddname of the SQL catalog index area
idms.syssql.ddlcatx	Dataset name of the SQL catalog index area
userdb	Ddname of the user database file
user.userdb	Dataset name of the user database file

If you submit utility statements: For information on the file assignments specific to utility statements, refer to the *CA-IDMS Utilities* document.

Executing in local mode: To specify that an IDMSBCF job stream is executing in local mode, remove the SYSCTL DD statement from the job stream and replace it with the following DD statements, as needed:

```
//dclscr   DD   DSN=idms.sysloc.ddlocscr,DISP=SHR
//secdd    DD   DSN=idms.sysuser.ddlsec,DISP=SHR
//dictdb   DD   DSN=idms.appldict.ddldml,DISP=SHR
//dloddb   DD   DSN=idms.appldict.ddldclod,DISP=SHR
//sysjrnl  DD   DSN=idms.tapejrnl,DISP=SHR
```

`dclscr`	Ddname of the local mode scratch (DDLOCSCR) area
`idms.sysloc.ddlocscr`	Dataset name of the local mode scratch (DDLOCSCR) area
`secdd`	Ddname of the system user catalog (DDLSEC) area
`idms.sysuser.ddlsec`	Dataset name of the system user catalog (DDLSEC) area
`dictdb`	Ddname of the application dictionary definition (DDLDML) area
`idms.appldict.ddldml`	Dataset name of the application dictionary definition (DDLDML) area
`dloddb`	Ddname of the application dictionary definition load (DDLDCLOD) area
`idms.appldict.ddldclod`	Dataset name of the application dictionary definition load (DDLDCLOD) area
`sysjrnl`	Ddname of the tape journal file
`idms.tapejrnl`	Dataset name of the tape journal file

Output file attributes: If you are putting the SYSLST output from IDMSBCF to a file, the attributes of that file must be as follows:

- RECFM = FB
- LRECL = 133
- BLKSIZE = 133*n

2.3 Commands for VM/ESA

Executing under the central version: Use the following VM/ESA commands for an IDMSBCF session that will run under the central version.

IDMSBCF (VM/ESA)

```
FILEDEF SYSIPT DISK sysipt data a (RECFM F LRECL ppp BLKSIZE ppp
EXEC IDMSFD
OSRUN IDMSBCF
```

`sysipt data a`	Filename, type, and mode of the file containing source statements
`ppp`	Record length of the input data file
`nnn`	Block size of the input data file

IDMSFD exec: The IDMSFD exec defines all FILEDEFs, TXTLIBs, and LOADLIBs required by the system.

Runtime parameters: IDMSFD references SYSIDMS, a file in which you can specify runtime parameters (such as dictionary and DMCL).

►►See the *CA-IDMS Database Administration*, for more information on SYSIDMS parameters.

Executing in local mode: To specify that IDMSBCF is executing in local mode, do one of the following:

- Link IDMSBCF with an IDMSOPTI program that specifies local execution mode
- Specify *LOCAL* as the first input parameter of the file identified in the FILEDEF SYSIPT statement
- Modify the OSRUN statement, as follows:

  ```
  OSRUN IDMSBCF PARM='*LOCAL*'
  ```

  Note: This option is valid only if you issue the OSRUN command from a System Product interpreter or an EXEC2 file.

Creating the SYSIPT file: To create the SYSIPT file, enter these VM/ESA commands:

```
XEDIT sysipt data a (NOPROF
INPUT
 .
 .
 .
source statements
 .
 .
 .
FILE
```

2.4 JCL for VSE/ESA

Executing under the central version: Use the following VSE/ESA JCL for an IDMSBCF session that will run under the central version.

IDMSBCF (VSE/ESA)

```
// EXEC PROC=IDMSLBLS
// UPSI b
// DLBL      idmspch,'temp.bcf',0
// EXTENT    SYS020,nnnnnn,,,ssss,llll
// ASSGN     SYS020,DISK,VOL=nnnnnn,SHR
// EXEC IDMSBCF
Source statements
/*
```

b	Appropriate UPSI switch, 1 through 8 characters, if specified in the IDMSOPTI module
idms.pch	File name of the dataset output from IDMSBCF
temp.bcf	File ID of the dataset output from IDMSBCF
SYS020	Logical unit assignment of the output
nnnnnn	Volume serial identifier of appropriate disk volume
ssss	Starting track (CKD) or block (FBA) of disk extent
llll	Number of tracks (CKD) or blocks (FBA) of disk extent

Runtime parameters: The procedure IDMSLBLS references SYSIDMS, a parameters file that allows you to specify physical requirements of your environment (DMCL, dictionary), runtime directives, and operating system-dependent file information.

▸▸See the *CA-IDMS Database Administration*, for more information on SYSIDMS parameters.

Executing in local mode: To execute IMSBCF in local mode, remove the UPSI specification and add the following statements:

```
// TLBL   sysjrnl,'idms.tapejrnl',,nnnnnn,,f
// ASSGN  SYS009,TAPE,VOL=nnnnnn
```

sysjrnl	Name of the tape journal file
idms.tapejrnl	ID of the tape journal file
f	File number of the tape journal file

`SYS009`	Logical unit assignment for the journal file

IDMSLBLS procedure: IDMSLBLS is a procedure that contains file definitions for the dictionaries, sample databases, and disk journal files provided during installation.

You can tailor the following IDMSLBLS procedure (provided on the installation tape) to reflect the filenames and definitions in use at your site. Reference IDMSLBLS as shown in the previous VSE/ESA JCL job stream.

```
* -------- LIBDEFS --------
// LIBDEF  *,SEARCH=idmslib.sublib
// LIBDEF  *,CATALOG=user.sublib
/*  ------------------------- LABELS -------------------------
// DLBL    idmslib,'idms.library',2099/365
// EXTENT  ,nnnnnn,,,ssss,1500
// DLBL    dccat,'idms.system.dccat',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,31
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    dccatl,'idms.system.dccatlod',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,6
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    dccatx,'idms.system.dccatx',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,11
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    dcdml,'idms.system.ddldml',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,101
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    dclod,'idms.system.ddldclod',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,21
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    dclog,'idms.system.ddldclog',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,401
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    dcrun,'idms.system.ddldcrun',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,68
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    dcscr,'idms.system.ddldcscr',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,135
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    dcmsg,'idms.sysmsg.ddldcmsg',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,201
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    dclscr,'idms.sysloc.ddlocscr',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,6
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    dirldb,'idms.sysdirl.ddldml',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,201
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    dirllod,'idms.sysdirl.ddldclod',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,2
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    empdemo,'idms.empdemo1',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,11
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    insdemo,'idms.insdemo1',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,6
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
```

```
// DLBL    orgdemo,'idms.orgdemo1',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,6
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    empldem,'idms.sqldemo.empldemo',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,11
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    infodem,'idms.sqldemo.infodemo',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,6
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    projdem,'idms.projseg.projdemo',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,6
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    indxdem,'idms.sqldemo.indxdemo',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,6
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    sysctl,'idms.sysctl',2099/365,SD
// EXTENT  SYSnnn,nnnnnn,,,ssss,2
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    secdd,'idms.sysuser.ddlsec',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,26
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    dictdb,'idms.appldict.ddldml',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,51
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    dloddb,'idms.appldict.ddldclod',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,51
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    sqldd,'idms.syssql.ddlcat',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,101
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    sqllod,'idms.syssql.ddlcatl',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,51
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    sqlxdd,'idms.syssql.ddlcatx',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,26
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    asfdml,'idms.asfdict.ddldml',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,201
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    asflod,'idms.asfdict.asflod',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,401
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    asfdata,'idms.asfdict.asfdata',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,201
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    ASFDEFN,'idms.asfdict.asfdefn',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,101
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
```

```
// DLBL    j1jrnl,'idms.j1jrnl',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,54
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    j2jrnl,'idms.j2jrnl',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,54
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    j3jrnl,'idms.j3jrnl',2099/365,DA
// EXTENT  SYSnnn,nnnnnn,,,ssss,54
// ASSGN   SYSnnn,DISK,VOL=nnnnnn,SHR
// DLBL    SYSIDMS,'#SYSIPT',0,SD
/+
/*
```

idmslib.sublib	Name of the sublibrary within the library containing CA-IDMS modules
user.sublib	Name of the sublibrary within the library containing user modules
idmslib	Name of the file containing CA-IDMS modules
idms.library	ID associated with the file containing CA-IDMS modules
SYS**nnn**	Logical unit of the volume for which the extent is effective
nnnnnn	Volume serial identifier of appropriate disk volume
ssss	Starting track (CKD) or block (FBA) of disk extent
dccat	Filename of the system dictionary catalog (DDLCAT) area
idms.system.dccat	ID of the system dictionary catalog (DDLCAT) area
dccatl	Filename of the system dictionary catalog load (DDLCATLOD) area
idms.system.dccatlod	ID of the system dictionary catalog load (DDLCATLOD) area
dccatx	Name of the system dictionary catalog index (DDLCATX) area
idms.system.dccatx	ID of the system dictionary catalog index (DDLCATX) area
dcdml	Name of the system dictionary definition (DDLDML) area
idms.system.ddldml	ID of the system dictionary definition (DDLDML) area
dclod	Name of the system dictionary definition load (DDLDCLOD) area
idms.system.ddldclod	ID of the system dictionary definition load (DDLDCLOD) area

dclog	Name of the system log area (DDLDCLOG) area
idms.system.ddldclog	ID of the system log (DDLDCLOG) area
dcrun	Name of the system queue (DDLDCRUN) area
idms.system.ddldcrun	ID of the system queue (DDLDCRUN) area
dcscr	Name of the system scratch (DDLDCSCR) area
idms.system.ddldcscr	ID of the system scratch (DDLDCSCR) area
dcmsg	Name of the system message (DDLDCMSG) area
idms.sysmsg.ddldcmsg	ID of the system message (DDLDCMSG) area
dclscr	Name of the local mode system scratch (DDLOCSCR) area
idms.sysloc.ddlocscr	ID of the local mode system scratch (DDLOCSCR) area
dirldb	Name of the IDMSDIRL definition (DDLDML) area
idms.sysdirl.ddldml	ID of the IDMSDIRL definition (DDLDML) area
dirllod	Name of the IDMSDIRL definition load (DDLDCLOD) area
idms.sysdirl.dirllod	ID of the IDMSDIRL definition load (DDLDCLOD) area
empdemo	Name of the EMPDEMO area
idms.empdemo1	ID of the EMPDEMO area
insdemo	Name of the INSDEMO area
idms.insdemo1	ID of the INSDEMO area
orgdemo	Name of the ORGDEMO area
idms.orgdemo1	ID of the ORDDEMO area
empldem	Name of the EMPLDEMO area
idms.sqldemo.empldemo	ID of the EMPLDEMO area
infodem	Name of the INFODEMO area
idms.sqldemo.infodemo	ID of the INFODEMO area
projdem	Name of the PROJDEMO area
idms.projseg.projdemo	ID of the PROJDEMO area
indxdem	Name of the INDXDEMO area
idms.sqldemo.indxdemo	ID of the INDXDEMO area
sysctl	Name of the SYSCTL file

idms.sysctl	ID of the SYSCTL file
secdd	Name of the system user catalog (DDLSEC) area
idms.sysuser.ddlsec	ID of the system user catalog (DDLSEC) area
dictdb	Name of the application dictionary definition area
idms.appldict.ddldml	ID of the application dictionary definition (DDLDML) area
dloddb	Name of the application dictionary definition load area
idms.appldict.ddldclod	ID of the application dictionary definition load (DDLDCLOD) area
sqldd	Name of the SQL catalog (DDLCAT) area
idms.syssql.ddlcat	ID of the SQL catalog (DDLCAT) area
sqllod	Name of the SQL catalog load (DDLCATL) area
idms.syssql.ddlcatl	ID of SQL catalog load (DDLCATL) area
sqlxdd	Name of the SQL catalog index (DDLCATX) area
idms.syssql.ddlcatx	ID of the SQL catalog index (DDLCATX) area
asfdml	Name of the asf dictionary definition (DDLDML) area
idms.asfdict.ddldml	ID of the asf dictionary definition (DDLDML) area
asflod	Name of the asf dictionary definition load (ASFLOD) area
idms.asfdict.asflod	ID of the asf dictionary definition load (ASFLOD) area
asfdata	Name of the asf data (ASFDATA) area
idms.asfdict.asfdata	ID of the asf data area (ASFDATA) area
ASFDEFN	Name of the asf data definition (ASFDEFN) area
idms.asfdict.asfdefn	ID of the asf data definition area (ASFDEFN) area
j1jrnl	Name of the first disk journal file
idms.j1jrnl	ID of the first disk journal file
j2jrnl	Name of the second disk journal file
idms.j2jrnl	ID of the second disk journal file
j3jrnl	Name of the third disk journal file
idms.j3jrnl	ID of the third disk journal file
SYSIDMS	Name of the SYSIDMS parameter file

2.5 JCL for BS2000/OSD

Executing under the central version: Use the following BS2000/OSD JCL for an IDMSBCF session that will run under the central version.

IDMSBCF (BS2000/OSD)

```
/ADD-FILE-LINK L-NAME=CDMSLIB,F-NAME=idms.dba.loadlib
/ADD-FILE-LINK L-NAME=CDMSLIB1,F-NAME=idms.loadlib
/ADD-FILE-LINK L-NAME=CDMSLODR,F-NAME=idms.loadlib
/ADD-FILE-LINK L-NAME=sysctl,F-NAME=idms.sysctl,SHARED-UPD=*YES
/ADD-FILE-LINK L-NAME=SYSIDMS,F-NAME=idms.sysidms
/ASSIGN-SYSDTA TO=*SYSCMD
/START-PROG *MOD(ELEM=IDMSBCF,LIB=idms.loadlib,RUN-MODE=*ADV)
statements you want to submit
```

`idms.dba.loadlib`	Filename of the load library containing the DMCL and database name table load modules
`idms.loadlib`	Filename of the load library containing CA-IDMS executable modules
`sysctl`	Linkname of the SYSCTL file
`idms.sysctl`	Filename of the SYSCTL file An IDMSOPTI module link edited with IDMSBCF can be used in place of or in addition to the SYSCTL file.
`idms.sysidms`	Filename of the SYSIDMS parameter file; use SYSIDMS to specify physical requirements of the environment (such as DMCL, dictionary name), runtime directives, and operating system-dependent file information ▸▸See the *CA-IDMS Database Administration*, for more information on SYSIDMS parameters.

If you submit a PUNCH statement: If you are going to submit any PUNCH statements, include this statement in JCL:

```
/ASSIGN-SYSOPT TO=#punch
```

Note: The resulting object needs to be put in an object library using the BS2KOBJM utility. The use of BS2KOBJM is described in *CA-IDMS Usage under BS2000/OSD*.

Executing in local mode: To execute IDMSBCF in local mode, remove SYSCTL and insert the following statements before the EXEC statement:

```
/ADD-FILE-LINK L-NAME=sysjrnl,F-NAME=idms.tapejrnl
/ADD-FILE-LINK L-NAME=dcdml,F-NAME=idms.system.ddldml,SHARED-UPD=*YES
/ADD-FILE-LINK L-NAME=dclod,F-NAME=idms.system.ddldclod,SHARED-UPD=*YES
/ADD-FILE-LINK L-NAME=dcmsg,F-NAME=idms.sysmsg.ddldcmsg,SHARED-UPD=*YES
```

`sysjrnl`	Linkname of the tape journal file
`idms.tapejrnl`	Filename of the tape journal file
`idms.system.ddldml`	Filename of the system dictionary definition area (DDLDML)
`idms.system.ddldclod`	Filename of the system dictionary definition load area (DDLDCLOD)
`idms.sysmsg.ddldcmsg`	Filename of the system message area (DDLDCMSG)
`dcdml`	Linkname of the system dictionary definition area (DDLDML)
`dclod`	Linkname of the system dictionary definition load area (DDLDCLOD)
`dcmsg`	Linkname of the system message area (DDLDCMSG)

Chapter 3. Online Processing

3.1 Overview of OCF 3-4
3.2 Beginning an OCF session 3-5
3.2.1 OCF screen format 3-5
3.3 Conducting an online session 3-7
3.4 Ending an OCF session 3-8
3.5 Executing statements dynamically 3-9
3.6 Executing statements stored in modules (OCFX) 3-10
3.6.1 Executing at the system prompt 3-10
3.7 Executing as part of a command list 3-12
3.7.1 OCFX status codes 3-12
3.8 Creating, editing, and saving modules 3-14
3.9 Commands for session control 3-15
3.9.1 Summary of session-control commands 3-15
3.10 Function keys assigned to session-control commands 3-17
3.10.1 APPLY 3-18
3.10.2 CLEAR 3-18
3.10.3 DISPLAY LINE 3-18
3.10.4 DISPLAY PAGE 3-19
3.10.5 END 3-20
3.10.6 ENTER= 3-20
3.10.7 ESCAPE 3-21
3.10.8 FIND 3-21
3.10.9 HELP 3-21
3.10.10 INSERT 3-22
3.10.11 PRINT 3-22
3.10.12 REPEAT 3-23
3.10.13 RESHOW 3-23
3.10.14 SUSPEND 3-23
3.10.15 SWAP 3-24
3.10.16 SWITCH 3-24
3.10.17 UPDATE 3-25
3.11 Commands for editing 3-26

OCF is the online component of the command facility. This chapter provides information on these OCF topics:

- A description of an OCF session
- Commands
- Saving and editing modules
- Executing statements at the system prompt and as part of a command list

3.1 Overview of OCF

This section describes the following:

- Beginning an OCF session
- OCF screen format
- Conducting an online session
- Ending an OCF session
- Session-control commands and function keys assigned to them
- Editing commands

3.2 Beginning an OCF session

To begin an online session:

1. Sign on to the host teleprocessing (TP) monitor, according to site-standard procedures
2. Enter OCF after the system prompt, as shown, then press [Enter]:

   ```
   ENTER NEXT TASK CODE: OCF [Enter]
   ```

3. Optionally, enter a CONNECT statement to establish a dictionary to which to connect during the session (see 1.4, “Connecting to a dictionary” on page 1-8)
4. Optionally, enter a SET OPTIONS statement to establish session-specific processing and output formatting options (see 1.5, “Using SET OPTIONS to select options” on page 1-9)

3.2.1 OCF screen format

OCF uses a standard screen that has:

- **A preformatted top line**
- **An unformatted input/output area**

```
                    OCF 15.0 ONLINE IDMS PAGE 1 LINE 1  DICT=SYSTEM     1/303

CONNECT TO DEMODICT;
*+ Status = 0
SET SESSION CURRENT SCHEMA DEMOEMPL;
*+ Status = 0
SELECT JOB_ID, JOB_TITLE, MIN_RATE, MAX_RATE
   FROM JOB
   WHERE MAX_RATE >= 27000.00
      ORDER BY JOB_TITLE;

*+ JOB_ID  JOB_TITLE                 MIN_RATE       MAX_RATE
*+ ------  ---------                 --------       --------
*+ 4023  Accountant                22000.00       60000.00
*+ 5890  Appraisal Spec            22500.00       35000.00
*+ 4130  Benefits Analyst          17500.00       28000.00
*+ 5110  CUST SER MGR              20000.00       54000.00
*+ 5111  CUST SER REP              13500.00       27000.00
*+ 6011  Manager - Acctng          29700.00       60500.00
*+ 6004  Manager - HR              33000.00       69000.00
*+ 6021  Manager - Mktng           38000.00       75000.00
*+ 4734  Mktng Admin               12500.00       31000.00
   •
   •
   •
*+ 25 rows processed
```

Top line: The top preformatted line of the screen contains the following areas:

1. **Command area** — Provides twenty spaces in columns 2 through 21 for entering commands that manipulate the work file or communicate with OCF; these

commands are listed under 3.9, “Commands for session control” on page 3-15 later in this chapter.

2. **Name area** — Displays the OCF header and the release number.
3. **Message area** — Displays one of the following, as appropriate: the work-file page and line number; the literal NO ERRORS; the number of error messages issued for the last update; or a message describing the status of a session-control command.
4. **Dictionary area** — Displays the current dictionary if you used the system DCUF SET DICTNAME statement or a CONNECT TO statement.
5. **Line number area** — Displays the current line of the screen I/O area, followed by the total number of lines (last line) in the work file.

Input/output area: Below the screen's top line, the input/output area covers the remainder of the screen. The default page width for OCF is 79 for input (76 for output because of the *+ and space characters preceding output lines).

3.3 Conducting an online session

After you have signed on to OCF, you can begin the online OCF session. Any physical DDL, logical DDL, SQL DML, command facility or security administration statements you enter on the OCF screen become part of your work file.

Work file options: Work file options are as follows:

- Enter and execute statements in the work file dynamically (see 3.5, "Executing statements dynamically" on page 3-9)
- Save the work file as a dictionary module (see 3.8, "Creating, editing, and saving modules" on page 3-14)
- Display the contents of any dictionary module for editing (see 3.8, "Creating, editing, and saving modules" on page 3-14)

Session options: You can tailor your processing and output options for the session using the SET OPTIONS statement (see 1.5, "Using SET OPTIONS to select options" on page 1-9). You can use the SET OPTIONS statement at any time during the session.

You can also enter commands on the top line of the screen to control an OCF session (see 3.9, "Commands for session control" on page 3-15).

Editing options: You can use line-editing commands to edit the statements contained in your OCF work file (see 3.11, "Commands for editing" on page 3-26).

Coding considerations: For information on coding considerations, see 1.2, "Coding considerations" on page 1-6.

Input and output are displayed: OCF displays each input statement, followed by the requested output. Output (including messages) is preceded by ***+**, indicating the text is commentary only; if you reexecute the statements on the screen, OCF ignores the text preceded by *+.

Note: To turn off the *+ commenting, specify WIDTH PAGE 79 in the SET OPTIONS statement.

Error handling: OCF responds to errors encountered in source input statements by listing status messages on the line immediately following the line in error.

Note: The message *+ Status = 0 indicates successful processing.

3.4 Ending an OCF session

To end an OCF session, choose one of the following options:

- **Enter SIGNOFF, LOGOFF, BYE, or END on the first line of the screen input area and press [Enter]**.

  This action:

  - Terminates the text editor
  - Deletes the contents of the work file
  - Clears the default processing options established for the session
  - Displays session statistics

  Press [Clear] to exit this screen and return control to the system.

- **Enter END in the command area and press [Enter].**

  Note: No session statistics are returned when you use END in the command area.

3.5 Executing statements dynamically

Enter statements in your work file according to the coding considerations noted in this document. Press [Enter] to execute the statements and display output.

To redisplay your input (without the output) for editing, press [PF9]. Edit the input and press [Enter] to execute the edited statements.

3.6 Executing statements stored in modules (OCFX)

You can use the OCFX statement to execute statements contained in modules with the language OCF. There are two methods for execution:

- Execution at the system prompt
- Execution as part of a system command list (CLIST)

This section describes both methods and provides a table listing and describing OCFX status codes.

3.6.1 Executing at the system prompt

To execute OCF-language modules at the system prompt, you enter the OCFX statement after the prompt. For example:

```
ENTER NEXT TASK CODE: OCFX ddlmod01 version 3 [Enter]
```

Syntax: OCFX statement

```
►►── OCFX module-name ──┬──────────────────────────┬─┬──────────┬──────────►◄
                        └─ Version version-number ─┘ ├─ Echo ← ─┤
                                                     └─ Noecho ─┘
```

Parameters:

OCFX
Executes the statements in an command facility source module, produces output, and returns control to the system (the ENTER NEXT TASK CODE prompt returns).

module-name
Specifies a specific command facility source module which contains statements you want to execute. The module must have a LANGUAGE attribute of OCF.

►► For information on the LANGUAGE parameter of the MODULE statement, refer to the *CA-IDMS IDD DDDL Reference Guide*.

Version *version-number*
Specifies a specific version for a load module to be used in an input session. The default version number is 1.

Echo
Specifies that the input statements are to be displayed on the screen before the output is displayed.

Noecho
Specifies that the named module's input statements are not to be displayed. If you use this parameter and also use SET OPTIONS STATUS OFF (see 1.5, “Using SET OPTIONS to select options” on page 1-9) as the first statement in the module's source code, then the screen displays only *non-zero* status messages.

Usage: OCFX produces output and then returns control to the system. If output appears on more than one screen, a message in the following format appears at the bottom of each screen:

```
PAGE nnn - NEXT PAGE:
```

Nnn is a variable representing the number of the current screen. You can page forward and backward through the screens as follows:

- Press [Enter] or [PA1] to page to the following screen.
- Press [PA2] to page to the previous screen.
- Type a page number after the NEXT PAGE prompt to select a specific screen, then press [Enter] to page to that screen.

3.7 Executing as part of a command list

To execute OCF-language modules as part of a system command list (CLIST):

1. Include the OCFX statement as one of the statements in the command list
2. Execute the command list

Note: For information on CLIST processing, refer to *CA-IDMS System Tasks and Operator Commands*.

3.7.1 OCFX status codes

The following table lists the status codes returned by OCFX processing.

Status code	Message and description
DC394001	**INVALID SYNTAX TOKEN FOUND** OCFX calls RHDCMISC/SCAN to parse input. When an improper token is encountered, OCFX abends.
DC394002	**INVALID MAJOR SCAN CODE** *<nnn>* OCFX calls RHDCMISC/SCAN to parse input. When an unexpected major scan code is returned, OCFX abends.
DC394003	**COMMAND EXCEEDS AVAILABLE SPACE** OCFX calls RHDCMISC/SCAN to parse input. Token values are extracted and kept in a command buffer which is returned to OCFX. The SCAN routine requires more space.
DC394004	**INVALID DICTIONARY/NODE NAME** OCFX extracts the dictionary name and node name from the current session PROFILE. It attempts to extract the command facility module with commands to be executed from this dictionary/node combination. The BIND failed.
DC394005	**INTERNAL STORAGE EXCEEDED** OCFX counts the number of TEXT-088 records in the dictionary module to be executed to determine how much storage to acquire for the command buffer. If this size is incorrect, the error results.
DC394006	**MODULE NOT FOUND** OCFX attempts to locate a specific command facility module name with either a specified or default (1) version number. The module must have a language attribute of OCF. If any of these conditions fail, the module cannot be loaded.

Status code	Message and description
DC394007	**MODULE HAS NO TEXT** OCFX located the command facility module name with either a specified or default (1) version number with a language attribute of OCF. However, the module has no TEXT-088 records.
DC394008	**UNEXPECTED DBMS ERROR** OCFX encountered a DBMS error which it does not have code to handle.

▸▸ For information on other status codes, refer to the *CA-IDMS Messages and Codes* document.

3.8 Creating, editing, and saving modules

OCF allows you to create, save, and edit OCF-language modules.

▸▸ For information on executing these modules, see 3.6, “Executing statements stored in modules (OCFX)” on page 3-10.

Creating and saving: To create a module that has the language OCF, do the following:

1. Enter source statements in your OCF work file
2. Enter the SAVE command on any line in the work file. Everything after the command to the end of the work file (or to a /* statement, whichever comes first) is saved in a module having the language OCF. Even input lines beginning with comment characters (*+) are saved.

```
▸▸── SAVE module-name ─┬──────────────────────┬─┬──────┬──────────────────▸◂
                      └─ Version version-number ─┘ └─ ; ─┘
```

Editing: To retrieve the contents of an OCF-language module for editing, use the EDIT command.

```
▸▸── EDIT module-name ─┬──────────────────────┬─┬──────┬──────────────────▸◂
                      └─ Version version-number ─┘ └─ ; ─┘
```

The contents of the module are displayed as an OCF work file. Edit the file using the editing commands shown in 3.11, “Commands for editing” on page 3-26.

Note: If you don't specify a version, the default is the version number set up in the command facility.

3.9 Commands for session control

Session-control commands direct an online OCF session. You can enter these commands in the command area (the top header line) of the screen.

▸▸ For information on editing commands that you enter in the input/output area of the screen, see 3.11, “Commands for editing” on page 3-26.

Entering session-control commands: To enter a session-control command, either type the command on the top line of the screen and press [Enter] or use the program function (PF) key assigned to the desired function (see 3.10, “Function keys assigned to session-control commands” on page 3-17).

You can abbreviate session-control commands to three characters, except for:

- FIND which can be abbreviated to F
- PRIOR which can be abbreviated to PRIO (four characters distinguish it from the keyword PRINT)

3.9.1 Summary of session-control commands

The following table gives short descriptions of the OCF session-control commands described in this section.

Command	Description
APPLY	Updates the screen and work file but does not execute the statements in the work file
CLEAR	Deletes all data contained in the work file
DISPLAY LINE	Displays a page of the work file, starting with the specified line
DISPLAY PAGE	Displays the requested page from the work file
END	Immediately terminates the current session
ENTER=APPLY ENTER=UPDATE	Sets the ENTER key to execute either the APPLY or the UPDATE command.
ESCAPE	Establishes the escape character that must be used with line editing commands
FIND	Locates a character string by searching forward or backward in the work file
HELP	Lists each session-control command and the associated PF key currently assigned to execute that command

Command	Description
INSERT	Inserts lines into the work file after the line at which the cursor is positioned
PRINT	Prints the contents of the work file
REPEAT	Repeats the line at which the cursor is positioned
RESHOW	Cancels all changes made to the current screen and redisplays the previous screen
SUSPEND	Suspends the current session and returns control to the host TP monitor
SWAP	Restores the screen and the work file to their condition prior to the last OCF execution
SWITCH (only if OCF is executing under the control of the Transfer Control Facility)	Suspends the session and transfers control to the specified online CA-IDMS component or to the Selection screen of the Transfer Control Facility
UPDATE	Updates the work file and executes the statements in the work file

3.10 Function keys assigned to session-control commands

Program function (PF) keys can be used as an alternative to typing session-control commands. To display the current PF-key assignments, use the HELP command.

The following table lists the PF keys established as OCF installation defaults.

PF key	Corresponding online command and description
PF1, PF8, PF13	DISPLAY PAGE NEXT Scrolls forward one page
PF2, PF7, PF14	DISPLAY PAGE PRIOR Scrolls backward one page
PF3, PF15	DISPLAY LINE NEXT Scrolls forward one line
PF4, PF16	INSERT Inserts up to a full screen of lines
PF5, PF17, [Enter]	APPLY Updates screen contents and work file but does not execute the commands in the work file
PF6, PF18, [Enter]	UPDATE Updates work file and executes the contents of the work file
PF9, PF21	SWAP Redisplays the input statements in the work file
PF12, PF24	PRINT Prints the contents of the work file (DC/UCF only)
PA1	Cancel FIND Cancels the FIND command
PA2	RESHOW Cancels changes to the current screen and redisplays the screen
CLEAR	CLEAR (CLR) Clears the contents of the work file
[Enter]	Updates the screen and work file (if you entered the command ENTER=APPLY since the start of your online session)

PF key	Corresponding online command and description
[Enter]	Updates the work file and executes the statements contained in the work file (if you entered the command ENTER=UPDATE since the start of your online session)

3.10.1 APPLY

The APPLY command updates the screen and work file without executing the statements in the work file. This command is useful for reviewing edits before executing the commands in the work file.

Syntax

```
►►── APPly ──────────────────────────────────────────────────────►◄
```

Example: An example of APPLY command usage follows:

1. Type INSERT. The text editor displays one line of existing text at the top of the I/O area, leaving the remainder of the area blank for entry of new lines.
2. Enter new lines of text.
3. Issue the APPLY command to add your new lines to the existing lines in the file. This allows you to review the file before executing the statements in the work file.
4. Make necessary corrections and issue the UPDATE command to execute the commands in the work file.

3.10.2 CLEAR

The CLEAR command deletes all lines contained in the work file. Lines cleared from the work file cannot be restored.

Syntax

```
►►──┬─ CLEar ─┬──────────────────────────────────────────────────►◄
    └─ CLR ───┘
```

3.10.3 DISPLAY LINE

The DISPLAY LINE command displays a specified line of the work file as the current work-file line. If the requested line is not the last line in the work file, subsequent lines are displayed beneath it.

Syntax

```
►►── DISplay Line ─┬─ FIRst ──────────┬──────────────────────────────────►◄
                   ├─ LASt ───────────┤
                   ├─ NEXt ───────────┤
                   ├─ PRIOr ──────────┤
                   ├─ line-number ────┤
                   ├─ *+ line-count ──┤
                   └─ *- line-count ──┘
```

FIRst
Displays the first line in the work file.

LASt
Displays the last line in the work file.

NEXt
Displays the line immediately following the first line in the screen I/O area.

PRIOr
Displays the line immediately preceding the first line in the screen I/O area.

line-number
Displays the specified line in the work file.

**+line-count*
Displays the line derived by adding *line-count* to the current work-file line number. For example, if the current work-file line number is 25, you display line 45 by specifying *+20.

**-line-count*
Displays the line derived by subtracting *line-count* from the current work-file line number. For example, if the current work file line number is 45, you display line 25 by specifying *-20.

3.10.4 DISPLAY PAGE

The DISPLAY PAGE command displays a specified page of the work file. A page is defined as the number of lines that the screen will accommodate, minus one for the top line.

You can request an explicit page number or a symbolic page number that indicates the relative position of the page within the work file.

Syntax

```
►►── DISplay PAGe ─┬─ FIRst ──────────┬──────────────────────────────────►◄
                   ├─ LASt ───────────┤
                   ├─ NEXt ───────────┤
                   ├─ PRIOr ──────────┤
                   ├─ page-number ────┤
                   ├─ *+ page-count ──┤
                   └─ *- page-count ──┘
```

FIRst
Displays the first page in the work file.

LASt
Displays the last page in the work file.

NEXt
Displays the page in the work file immediately following the current page.

PRIOr
Displays the page in the work file immediately preceding the current page.

page-number
Displays the specified page in the work file.

**+page-count*
Displays the page derived by adding *line-count* to the current work-file page number. For example, if the current work-file page number is 2, you display page 6 by specifying *+4.

**-page-count*
Displays the page derived by subtracting *page-count* from the current work-file page number. For example, if the current page is 6, you display page 4 by specifying *-2.

3.10.5 END

The END command terminates the online session, deletes the contents of the work file, and signs you off from OCF.

Syntax

```
►►── END ──────────────────────────────────────────────────────────────►◄
```

3.10.6 ENTER=

The ENTER command sets the function of the ENTER key to either APPLY or UPDATE for the duration of the session or until you issue another ENTER command.

Syntax

```
►►── ENTer= ─┬─ UPDate ─┬──────────────────────────────────────────────►◄
           └─ APPly ─┘
```

UPDate
Updates the work file and executes the commands in the work file. UPDATE is the default.

APPly
Updates the work file but *does not* execute the commands in the work file.

3.10.7 ESCAPE

The ESCAPE command establishes the escape character you must enter as the first character in a line editing command (see 3.11, “Commands for editing” on page 3-26).

When you specify a line editing command, you type the current escape character in column 1 of the work file. The escape character causes the following characters to be evaluated as a line editing command, rather than as an input statement.

Syntax

```
►►── ESCape escape-character ──────────────────────────────────────────►◄
```

Escape-character can be any alphanumeric character. The percent sign (%) is the default.

3.10.8 FIND

The FIND command locates a specified character string by searching the work file. FIND begins the search at the current page, displays the first page on which the character string is found, and highlights the lines that contain the object string:

- To continue the FIND operation, press [Enter].
- To discontinue the FIND operation before reaching the end (or beginning) of the work file, press [PA1].

Syntax

```
►►── Find 'character-string' ┬─ FWD ─┬──────────────────────────────────►◄
                             └─ BAC ─┘
```

Find '*character-string***'**
Initiates a search for the *character-string* in the direction specified by FWD or BACK. You must enclose *character-string* in single quotes.

FWD
Searches from the current line to the end of the work file. FWD is the default.

BAC
Searches from the current line to the beginning of the work file.

3.10.9 HELP

The HELP command displays a list of each session-control command and the PF key currently assigned to execute that command. To return to the work file, press [Enter].

Syntax

```
►►── HELp ──────────────────────────────────────────────────────────────►◄
```

3.10.10 INSERT

The INSERT command allows you to insert lines into the work file.

Syntax

```
►►── INSert ──────────────────────────────────────────────────────────►◄
```

How to use INSERT: During an online OCF session, perform the following steps to insert text into the work file:

1. Issue the INSERT command, using one of the following methods:
   - **Session-control command** — Enter INSERT in the command area, position the cursor at the line that the inserted lines will *follow*, and press [Enter].
   - **PF key** — Position the cursor at the line that the inserted lines will *follow*. Press the PF key currently assigned to execute the INSERT command (PF4 is the default).
2. Enter text on as many blank lines as necessary. The new text exists only on the terminal screen and has yet to be placed into the work file.
3. Continue issuing INSERT commands as each page is filled until all input is entered.
4. Merge the text with the work file by performing one of the following actions:
   - **APPLY** — To update the screen and the work file, enter the APPLY command, discussed earlier in this section. If ENTER=APPLY then you can press [Enter] and get the same results.

     The text editor inserts the new lines into the work file and updates the screen without executing the statements in the work file. To execute the statements, use the UPDATE command.
   - **UPDATE** — To execute the commands, enter the UPDATE command, discussed later in this section. If ENTER=UPDATE, then you can press [Enter] and get the same results.

     The text editor inserts the new lines into the work file and executes the commands in the updated work file.

3.10.11 PRINT

The PRINT command prints the contents of the work file on a hard-copy printer. The system acknowledges receipt of the command with a PRINT INITIATED message in the message area.

Syntax

```
►►── PRInt ──────────────────────────────────────────────────────────►◄
```

3.10.12 REPEAT

The REPEAT command repeats a work-file line a specified number of times. To issue a REPEAT command, enter REPEAT in the command area, position the cursor at the line to be repeated, and press [Enter].

Syntax

```
►►── REPeat repeat-count ──────────────────────────────────────────►◄
```

REPeat
Repeats the specified line once.

repeat-count
Represents the number of times the line is to be repeated.

The maximum value for *repeat-count* is two less than the number of lines the terminal screen will accommodate. For example, on a 24-line terminal the maximum for *repeat-count* is 22. Entering an integer larger than the maximum value causes an error condition; reissue the command.

3.10.13 RESHOW

The RESHOW command cancels all changes made to the current screen, then redisplays the current screen as of the last time a control key was pressed.

Syntax

```
►►── REShow ──────────────────────────────────────────────────────►◄
```

3.10.14 SUSPEND

The SUSPEND command allows you to exit from a session without ending the session. The current session is suspended, and control returns to the DC/UCF system. The contents of the work file and session options are saved. You can proceed to another online component, then return to the suspended session at the point at which you exited.

If you sign off from the DC/UCF system, or if the system is terminated while a session is suspended, the work file and all session options are lost.

When the session is operating under the Transfer Control Facility (TCF), SUSPEND is synonymous with SWITCH, described later in this section.

►► For information on the Transfer Control Facility, refer to the *CA-IDMS Transfer Control Facility* document.

Syntax

```
►►── SUSpend ──────────────────────────────────────────────────────►◄
```

3.10.15 SWAP

The SWAP command restores the screen and the work file to their condition prior to the last OCF execution. SWAP is a convenient means of editing OCF input. For example, if OCF returns error messages for your input, you can use SWAP to redisplay the original input. You can correct the input and then reexecute it.

Swapping screens does not revoke the results of the previous execution.

Syntax

```
►►── SWAp ──────────────────────────────────────────────────────────►◄
```

3.10.16 SWITCH

The SWITCH command is available only when you are using OCF under the Transfer Control Facility (TCF). TCF allows you to transfer from one development tool to another without having to sign off and return to the DC/UCF system.

With SWITCH, you can:

- Display the TCF Selection screen
- Transfer to a previously suspended session of an online development tool (like OCF), as long as that tool is defined in TCF
- Initiate a new session of an online development tool that is defined in TCF
- Terminate or suspend a TCF session and return to the DC/UCF system

Note: For information about TCF, refer to the *CA-IDMS Transfer Control Facility* document.

Syntax

```
►►── SWitch ─┬──────────────────────────────────┬──────────────────────►◄
             ├─ product-code ─┬─┬──────────┬─┤
             │└─ task-code ───┘ ├─ OLD ─┤  │
             │                  └─ NEW ─┘  │
             ├─ tcf-task-code ──────────────┤
             ├─ CLear ─┬──────────────────┤
             │└─ CLR ───┘                  │
             └─ SUSpend ────────────────────┘
```

product-code
Identifies a development tool by a unique system-supplied invocation name. For example, command facility is the product code for the online command facility compiler; OCF is the code for OCF.

task-code
Identifies a site-defined product code. For example, IDDT can be defined as the task code for the online command facility; SSCT can be the task code for the online subschema compiler. Task codes are defined at system generation and can vary from site to site.

OLD/NEW
Qualifies the *product-code/task-code* specification as follows:

- OLD restarts the most recently suspended session of the requested development tool. If no such session exists, a new session of the requested development tool is started.
- NEW starts a new session of the specified development tool. If you have a suspended OCF session, NEW resumes that suspended session. This is because you can only have one suspended session for each of these tools.

tcf-task-code
Requests the display of the TCF Selection Screen. The *tcf-task-code* for your site is specified at system generation time. The default task code is TCF.

The TCF Selection Screen provides the following information:

- A list of suspended sessions
- A list of all development tools available under TCF
- A list of database and data dictionary defaults (dbname, dictname, nodename, and dictnode)

CLEar
Suspends OCF, ends the TCF session (clearing the list of suspended sessions and defaults), and returns control to the DC/UCF system.

Suspended sessions exist until you sign off from the host TP monitor or the DC/UCF system terminates.

SUSpend
Suspends OCF, suspends the TCF session (saving the list of suspended sessions and defaults), and returns control to the DC/UCF system.

When TCF is next executed, the list is displayed in the suspended sessions area of the TCF Selection screen. The list of suspended sessions is saved by TCF when you sign off from the host TP monitor or the DC/UCF system terminates.

3.10.17 UPDATE

The UPDATE command applies your most recent changes to the work file *and* executes the commands in the work file.

Note: Use the APPLY command if you want to update the work file without executing the commands.

Syntax

```
►►── UPDate ──────────────────────────────────────────────────────────►◄
```

3.11 Commands for editing

You use line-editing commands to copy, delete, move, and repeat lines or blocks of lines within a work file.

How to enter a line-editing command: Begin a line-editing command with an *escape character* (%), and end the command with a space. Enter the command in column 1 of the line to which the command applies.

Note: When line-editing commands are present in your work file, pressing [Enter] does not invoke OCF (even if you used the ENTER=UPDATE command).

Command formats: The following table gives the names of the commands and the formats for entering them. Note that *n* is a variable for a value that you specify. *N* represents the number of lines the command should act on; it includes the line the command is on.

Edit command	Command format
COPY	Copy a single line: %C Copy this line and the next *n-1* lines: %C*n* Copy a block of lines: %CB (on the first line of the block) %CE (on the last line of the block) You follow a COPY command with an AFTER or BEFORE command.
DELETE	Delete a single line: %D Delete this line and the next *n-1* lines: %D*n* Delete a block of lines: %DB (on the first line of the block) %DE (on the last line of the block)

Edit command	Command format
MOVE	Move a single line: %M Move this line and the next *n-1* lines: %M*n* Move a block of lines: %MB (on the first line of the block) %ME (on the last line of the block) You follow a MOVE command with an AFTER or BEFORE command.
REPEAT	Repeat a single line: %R Repeat this line and the next *n-1* lines: %R*n* Repeat a block of lines: %RB (on the first line of the block) %RE (on the last line of the block)
AFTER	Place a COPY or MOVE block *after* the line on which this command is placed: %A
BEFORE	Place a COPY or MOVE block *before* the line on which this command is placed: %B
TOP	Reposition the work file so that the line on which the command appears becomes the top line on the screen: %T

Appendix A. Using the Command Facility as a Subprogram

A.1 Overview A-3
A.2 Compiler interface parameter list A-5
A.2.1 CIO block A-5
A.2.2 CIOF block A-6
A.2.3 User parameters A-7
A.3 Work-area file A-8
A.4 Sample program that calls the command facility A-9

A.1 Overview

Any program can call the command facility compiler (IDMSOCF) as a subroutine to extract information from or update information in the dictionary. The program or dialog passes to IDMSOCF an input file that contains the command facility statements to be used to obtain the desired information. The command facility compiler places the extracted data in an output file, which can be examined and processed by the program or dialog.

The command facility compiler uses these files:

- An input file (SYSIPT)
- A print file (SYSLST)
- A punch file (SYSPCH)

Each of these files consists of 80-byte records. Normally, the compiler controls these files, directing the input and print files to the terminal and discarding the punch file. However, when a program or dialog calls IDMSOCF as a subroutine, the calling program specifies that these three files can be directed to a work-area file, a scratch area, a queue, another program, or a null file. Advantages and disadvantages associated with each of these storage mechanisms are as shown in the following table.

Storage type	Advantages/disadvantages
Work-area file	Offers the fastest access but is limited in size; this mechanism is the best choice for small files.
Scratch area	Can accommodate a large volume of data; however, scratch areas are volatile and may require the calling program to perform I/Os.
Queue	Can accommodate a large volume of data; however, the calling program must perform I/Os and initiate run units to access queues.
User program exit	Offers the most advantages. The user has maximum control over the file's records, selecting certain records for special processing.
Null file	Suppresses the output from IDMSOCF. If the program tries to read the null file, an end-of-file condition is returned immediately.

Combining storage types: It may be advantageous to combine two mechanisms. For example, direct the file to a user program exit that directs a work area's overflow to a scratch area. The work area is described under A.3, “Work-area file” on page A-8.

Input file statements: The input file can contain any valid command facility statements. All standard compiler security applies to issuing these statements.

First statement — CONNECT: To ensure proper access to the compiler, it is recommended that the first command in the input file be CONNECT.

The dictionary named in the CONNECT command must identify the dictionary to be accessed.

How the compiler is called: To call the command facility compiler, the user program issues a LINK request, naming the module IDMSOCF and passing seven parameters: a compiler input/output (CIO) block, one compiler input/output file (CIOF) block for each of the three command facility files. (input, print, and punch), followed by a user parameter for each of the three files (input, print, and punch).

A.2 Compiler interface parameter list

The CIO block, CIOF block, and user parameters are described separately in this section.

A.2.1 CIO block

The CIO block contains return codes that indicate the presence of invalid CIO or CIOF parameters and specify the outcome of the RETURN operation from the compiler to the user program. This block also contains a value that indicates which compiler is in error. The CIO block is formatted as follows:

Field	Label	Usage	Size	Picture	Description
1	CIOID	Char	4	X(4)	Compiler I/O ID; must be initialized to 'CIO '
2	CIOUSER	Binary	4	`S9(8) COMP`	Reserved for user storage (normally, an address); must be initialized to 0
3	CIOIORC	Binary	4	`S9(8) COMP`	CIO return code
4	CIOCMPRC	Binary	4	`S9(8) COMP`	Compiler return code
5		Char	8	X(8)	Reserved
6	CIOERRFI	Char	8	X(8)	If an error has occurred, the name of the file in error

Error conditions are returned to the calling program as a return code in the CIOIORC field of the CIO block. These error codes are described as follows:

Error code	Description
4 (X'04')	An invalid parameter has been passed to the compiler interface; typically, the CIO or CIOF block has not been properly initialized.
8 (X'08')	The scratch id or queue id specified in the CIOF block could not be found.
28 (X'1C')	An I/O error has occurred during an attempt to access a scratch area or queue file, or an output work-area file is exhausted.

The name of the file in error (SYSIPT, SYSLST, or SYSPCH) is placed in the CIOERRFI field of the CIO block for examination by the user.

A.2.2 CIOF block

For each file, the user program passes a CIOF block. Each CIOF block describes the type of file being passed (for example, work area); the number of records to be read from and written to the file; and the maximum number of records the file can hold. The format of the CIOF block is as shown in the following table.

Field	Label	Usage	Size	Picture	Description
1	CIOFTYPE	Char	8	X(8)	File type for data block (see next table)
2	CIOFNAME	Char	16	X(16)	File name (see next table)
3	CIOFRC	Binary	4	`S9(8) COMP`	`Program return code, if file type is 'LINKPGM'/ 'LINKEPA'.`
4	CIOFZUS	Binary	4	`S9(8) COMP`	Number of records read from or written to file; updated if file type is 'WORKAREA', 'SCRATCH', or 'QUEUE'; can also be updated by program if file type is 'LINKPGM'/'LINKEPA'.
5	CIOFSZMX	Binary	4	`S9(8) COMP`	If file type is 'WORKAREA', maximum number of records that file holds (SYSIPT) or can hold (SYSLST or SYSPCH).

Fields 1 and 2: The following table provides guidelines for determining file types and names for fields 1 and 2 of the CIOF block.

File type	File name
For a work area initialize to 'WORKAREA'	Not applicable
For a scratch area initialize to 'SCRATCH '	Scratch area ID
For a queue initialize to 'QUEUE '	Queue ID
For a program initialize to 'LINKPGM ' or 'LINKEPA', based on the LINK mechanism used	If 'LINKPGM' use the 8-character program name If 'LINKEPA' use the 4-character entry point address
For a null file initialize to 'NULL'	Not applicable

A.2.3 User parameters

Each CIOF block must be followed by a user parameter that specifies the location of the work-area file within program variable storage or defines the parameter list to be passed to the user program exit. The user parameters and the information passed in each are as follows:

Parameter	Information passed
1	CIO block
2	CIOF block for input file
3	If CIOFTYPE='WORKAREA', the work area allocated in program variable storage; if CIOFTYPE='LINKPGM' or 'LINKEPA', parameter list to be passed to the user exit; otherwise, this should be the literal 'NULL'
4	CIOF block for the output print file
5	If CIOFTYPE='WORKAREA', the work area allocated in program variable storage; if CIOFTYPE='LINKPGM' or 'LINKEPA', the parameter list to be passed to the user exit; otherwise, this should be the literal 'NULL'
6	CIOF block for the output punch file
7	If CIOFTYPE='WORKAREA', the work area allocated in program variable storage; if CIOFTYPE='LINKPGM' or 'LINKEPA', the parameter list to be passed to the user exit; otherwise, this should be the literal 'NULL'

A.3 Work-area file

The work-area file is a block of program variable storage that contains a series of 80-byte records. The following rules apply to the work-area file:

- The maximum number of records in the work-area file must be placed in the CIOFSZMX field of the applicable CIOF block by the user program before the program invokes the compiler interface.
- The size of the work-area file is determined by the user program; it must be a multiple of 80.
- If an compiler output file is exhausted when the work-area file is written to by the compiler, a return code of 28 (X'1C') is placed in the CIOIORC field of the CIO block and the excess records are lost.
- If the compiler file is exhausted when the work-area file is read from by the compiler, an end-of-file condition is returned to the compiler.

Upon return to the user program, the CIOFSZUS field contains the number of records actually read from or written to the file.

A.4 Sample program that calls the command facility

The following sample COBOL program calls the command facility and requests the command facility to display an element.

```
************************************************************
 IDENTIFICATION DIVISION.
************************************************************
 PROGRAM-ID.            CALLIDD.
 DATE WRITTEN.          MARCH 1, 1991.
 DATE COMPILED.
************************************************************
* REMARKS.
************************************************************
*
*  THIS IS A SAMPLE DC COBOL PROGRAM THAT DEMONSTRATES HOW
*  AN APPLICATION PROGRAM CAN CALL
*  COMMAND FACILITY AS A SUBPROGRAM AND
*  PASS TO COMMAND FACILITY A REQUEST TO DISPLAY AN ELEMENT.
*  THE OUTPUT
*  OF THE REQUEST IS DISPLAYED BY THE COBOL PROGRAM.
*
************************************************************
 ENVIRONMENT DIVISION.
************************************************************
 CONFIGURATION SECTION.
 SOURCE-COMPUTER.       IBM-370.
 OBJECT-COMPUTER.       IBM-370.

 IDMS-CONTROL SECTION.
 PROTOCOL.              MODE IS IDMS-DC DEBUG
                        IDMS-RECORDS MANUAL.
************************************************************
 DATA DIVISION.
************************************************************
 MAP SECTION.
     MAP CDSIMAP1.
     EJECT
************************************************************
 WORKING-STORAGE SECTION.
************************************************************
 01  BEGIN-WS.
     03 FILLER                  PIC X(40) VALUE
        '***** WORKING STORAGE BEGINS HERE ******'.
************************************************************
*    SWITCHES-AREA - PROGRAM CONTROL SWITCHES              *
************************************************************
 01  SWITCHES-AREA.
     03 FILLER                  PIC X(08) VALUE 'SWITCHES'.
     03 COMMAND-FACILITY-EOF-SW              PIC 9 VALUE 0.
        88 COMMAND-FACILITY-EOF              VALUE 1.
     03 FIRST-TIME-SW           PIC 9 VALUE 0.
        88 FIRST-TIME           VALUE 1.
     03 ERROR-SW                PIC 9 VALUE 0.
        88 NO-ERRORS            VALUE 0.
```

```
*************************************************************
*    WORK-FIELDS   - PROGRAM WORK FIELDS                    *
*************************************************************
 01  WORK-FIELDS.
     03 FILLER                  PIC X(08) VALUE 'WORKAREA'.
     03 SUB                     PIC 99 VALUE 0.
     03 AID-BYTE                PIC X.
        88 CLEAR-HIT            VALUE '_'.
        88 PA1-HIT              VALUE '%'.
     03 TASK-CODE               PIC X(8).
     03 GOOD-RC                 PIC S9(8) COMP VALUE +0.
     03 Q-EL                    PIC X(32) VALUE ALL '?'.
*************************************************************
*    MESSAGES-AREA - OPERATOR MESSAGES                      *
*************************************************************
 01  MESSAGES-AREA.
     03 FILLER                  PIC X(08) VALUE 'MESSAGES'.
     03 OK-MSG                  PIC X(40) VALUE
        'PROCESSING COMPLETE - PROCEED          '.
     03 NO-ELEMENT-MSG          PIC X(40) VALUE
        'ELEMENT NAME MISSING, PLEASE FILL IT IN'.
     03 CIO-ERROR-MSG.
        05 FILLER               PIC X(36) VALUE
        'CIO PROCESSING ERROR - RETURN CODE ='.
        05 CIO-ERROR-CODE       PIC X(4) VALUE '0000'.
*************************************************************
*    SCR-RCD       - SCRATCH RECORD AREA                    *
*************************************************************
 01  SCR-RCD.
     03 SCR-DBK                 PIC S9(8) COMP.
     03 SCR-RCDID               PIC S9(8) COMP.
     03 SCR-STATUS              PIC X.
     03 SCR-RCD-END             PIC X.
     EJECT
*************************************************************
*    PARAMETER 1 - THE COMPILER INOUT/OUTPUT BLOCK          *
*************************************************************
 01  CIO-PARM1.
     03 CIO-ID                  PIC X(4)  VALUE 'CIO '.
     03 CIO-USER                PIC S9(8) COMP VALUE +0.
     03 CIO-IO-RC               PIC S9(8) COMP VALUE +0.
     03 CIO-COMMAND FACILITY-RC      PIC S9(8) COMP VALUE +0.
     03 CIO-RESERVED            PIC X(8)  VALUE SPACES.
     03 CIO-ERROR-FILE          PIC X(8)  VALUE SPACES.
        88 SYSIPT-ERROR         VALUE 'SYSIPT'.
        88 SYSLST-ERROR         VALUE 'SYSLST'.
        88 SYSPCH-ERROR         VALUE 'SYSPCH'.
     03 CIO-NULL                PIC X(4) VALUE 'NULL'.
*************************************************************
*    PARAMETER 2 - CIOF INPUT BLOCK                         *
*************************************************************
```

```
 01  CIO-PARM2.
     03 CIOF-I-TYPE              PIC X(8) VALUE 'WORKAREA'.
     03 CIOF-I-NAME              PIC X(16) VALUE SPACES.
     03 CIOF-I-F-RC              PIC S9(8) COMP VALUE +0.
     03 CIOF-I-SIZE-US           PIC S9(8) COMP VALUE +0.
     03 CIOF-I-SIZE-MAX          PIC S9(8) COMP VALUE +5.
*************************************************************
*    PARAMETER 3 - INPUT DATA AREA                          *
*************************************************************
 01  CIO-PARM3.
     03 CIO-I-LINE1.
        05 FILLER                PIC X(20) VALUE
           'CONNECT TO '.
        05 CIO-I-DICT            PIC X(8)  VALUE SPACES.
        05 FILLER                PIC X(52) VALUE ';'.
     03 CIO-I-LINE2.
        05 FILLER                PIC X(80) VALUE
           'SET SESSION READ ONLY;'.
     03 CIO-I-LINE3.
        05 FILLER                PIC X(20) VALUE
           'SET CURRENT SCHEMA '.
        05 CIO-I-SCHEMA          PIC X(18) VALUE SPACES.
        05 FILLER                PIC X(52) VALUE ';'.
     03 CIO-I-LINE4.
        05 FILLER                PIC X(20) VALUE
           'DISPLAY TABLE '.
        05 CIO-I-TABLE           PIC X(18) VALUE SPACES.
        05 FILLER                PIC X(52) VALUE ';'.
*************************************************************
*    PARAMETER 4 - CIOF OUTPUT BLOCK                        *
*************************************************************
 01  CIO-PARM4.
     03 CIOF-O-TYPE              PIC X(8) VALUE 'WORKAREA'.
     03 CIOF-O-NAME              PIC X(16) VALUE SPACES.
     03 CIOF-O-F-RC              PIC S9(8) COMP VALUE +0.
     03 CIOF-O-SIZE-US           PIC S9(8) COMP VALUE +0.
     03 CIOF-O-SIZE-MAX          PIC S9(8) COMP VALUE +100.
*************************************************************
*    PARAMETER 5 - OUTPUT DATA AREA                         *
*************************************************************
 01  CIO-PARM5.
     03 CIOF-OUTPUT-LINE         PIC X(80)
         OCCURS 100 TIMES.
     EJECT
*************************************************************
*    PARAMETER 6 - CIOF PUNCH BLOCK                         *
*************************************************************
 01  CIO-PARM6.
     03 CIOF-P-TYPE              PIC X(8) VALUE 'WORKAREA'.
     03 CIOF-P-NAME              PIC X(16) VALUE SPACES.
     03 CIOF-P-F-RC              PIC S9(8) COMP VALUE +0.
     03 CIOF-P-SIZE-US           PIC S9(8) COMP VALUE +0.
     03 CIOF-P-SIZE-MAX          PIC S9(8) COMP VALUE +0.
```

```
*************************************************************
*    PARAMETER 7 - PUNCH DATA AREA                          *
*************************************************************
 01  CIO-PARM7                  PIC X(80) VALUE 'NULL'.
*************************************************************
*    IDMS AREA                                              *
*************************************************************
 COPY IDMS SUBSCHEMA-CTRL.
 COPY IDMS MAP-CONTROLS.
 COPY IDMS MAP-RECORDS.
     EJECT
 PROCEDURE DIVISION.
*************************************************************
*                                                           *
* ROUTINE - 0000-MAIN-LINE                                  *
*                                                           *
* THIS ROUTINE IS THE MAIN CONTROL OF THE PROGRAM, CALLING  *
* THE OTHER ROUTINES TO DO THE ACTUAL WORK.                 *
*                                                           *
*************************************************************
 0000-MAIN-LINE.
     PERFORM 1000-GET-SCRATCH-REC THRU 1999-EXIT.
     IF FIRST-TIME
         PERFORM 2000-DISPLAY-MAP THRU 2999-EXIT
         GO TO 0800-RETURN-SCREEN.
     PERFORM 3000-GET-MAP THRU 3999-EXIT.
     IF CLEAR-HIT
         GO TO 0900-DC-RETURN.
     PERFORM 4000-EDIT-DATA THRU 4999-EXIT.
     IF NO-ERRORS
         PERFORM 5000-CALL-COMMAND FACILITY THRU 5999-EXIT.
     MAP OUT USING CDSIMAP1 WAIT IO OUTPUT DATA YES.

*************************************************************
*                                                           *
* ROUTINE - 0800-RETURN-SCREEN                              *
*                                                           *
* THIS ROUTINE SETS UP THE RETURN SO THAT THIS TRANSACTION  *
* WILL BE THE NEXT TRANSACTION EXECUTED FROM THE TERMINAL.  *
*                                                           *
*************************************************************
 0800-RETURN-SCREEN.
     ACCEPT TASK CODE INTO TASK-CODE.
     DC RETURN NEXT TASK CODE TASK-CODE.
```

```
*************************************************************
*                                                           *
* ROUTINE - 0900-DC-RETURN                                  *
*                                                           *
* THIS ROUTINE DELETES THE SCRATCH RECORD AND THEN RETURNS  *
* CONTROL TO THE DC SYSTEM.                                 *
*                                                           *
*************************************************************
 0900-DC-RETURN.
     DELETE SCRATCH RECORD ID SCR-RCDID.
     DC RETURN.
     EJECT
*************************************************************
*                                                           *
* ROUTINE - 1000-GET-SCRATCH-REC.                           *
*                                                           *
* THIS ROUTINE ATTEMPTS TO GET THE SCRATCH RECORD, WHICH    *
* IS USED TO DETERMINE IF THIS IS THE FIRST TIME THE        *
* TRANSACTION HAS BEEN EXECUTED.                            *
*                                                           *
*************************************************************
 1000-GET-SCRATCH-REC.
     MOVE 1 TO SCR-RCDID.
     GET SCRATCH RECORD ID SCR-RCDID KEEP
         INTO SCR-RCD TO SCR-RCD-END
         ON ANY-ERROR-STATUS
            IF ERROR-STATUS NOT = '0000'
                MOVE 1 TO FIRST-TIME-SW
            ELSE
                MOVE 0 TO FIRST-TIME-SW.
 1999-EXIT.
     EXIT.
*************************************************************
*                                                           *
* ROUTINE - 2000-DISPLAY-MAP                                *
*                                                           *
* THIS ROUTINE CREATES A SCRATCH RECORD AND DOES THE INITIAL *
* MAP OUT.                                                  *
*                                                           *
*************************************************************
 2000-DISPLAY-MAP.
     MOVE 0 TO SCR-DBK.
     MOVE '1' TO SCR-STATUS.
     PUT SCRATCH FROM SCR-RCD TO SCR-RCD-END
         RECORD ID SCR-RCDID.
     PERFORM 8000-INITILIZE-MAP THRU 8099-EXIT.
     MAP OUT USING CDSIMAP1 OUTPUT NEWPAGE.
 2999-EXIT.
     EXIT.
```

```
*************************************************************
*                                                           *
* ROUTINE - 3000-GET-MAP                                    *
*                                                           *
* THIS ROUTINE GETS THE MAP.                                *
*                                                           *
*************************************************************
 3000-GET-MAP.
     PERFORM 8000-INITILIZE-MAP THRU 8099-EXIT.
     MAP IN USING CDSIMAP1.
     INQUIRE MAP CDSIMAP1 MOVE AID TO AID-BYTE.
 3999-EXIT.
     EXIT.

     EJECT
*************************************************************
*                                                           *
* ROUTINE - 4000-EDIT-DATA                                  *
*                                                           *
* THIS ROUTINE CHECKS THE ELEMENT NAME TO SEE IF IT HAS BEEN *
* FILLED IN.  IF IT IS BLANK OR NULLS, AN ERROR MESSAGE IS   *
* DISPLAYED, AND THE MAP IS RETURNED TO THE OPERATOR FOR     *
* CORRECTION.                                               *
*                                                           *
*************************************************************
 4000-EDIT-DATA.
     MOVE 0 TO ERROR-SW.
     IF (CDSIDICT = SPACES)
     OR (CDSIDICT = LOW-VALUES)
         MOVE 1 TO ERROR-SW
         MOVE NO-ELEMENT-MSG TO CDSIMSG
         MOVE Q-EL TO CDSIELNM
         MODIFY MAP CDSIMAP1 TEMPORARY
             FOR CDSIELNM ATTRIBUTES BRIGHT
         GO TO 4999-EXIT.
     MOVE CDSIELNM TO CIO-I-DICT.
     MOVE 0 TO ERROR-SW.
     IF (CDSISCHEMA = SPACES)
     OR (CDSISCHEMA = LOW-VALUES)
         MOVE 1 TO ERROR-SW
         MOVE NO-ELEMENT-MSG TO CDSIMSG
         MOVE Q-EL TO CDSIELNM
         MODIFY MAP CDSIMAP1 TEMPORARY
             FOR CDSIELNM ATTRIBUTES BRIGHT
         GO TO 4999-EXIT.
     MOVE CDSIELNM TO CIO-I-SCHEMA.
```

```
     MOVE 0 TO ERROR-SW.
     IF (CDSITABLE = SPACES)
     OR (CDSITABLE = LOW-VALUES)
         MOVE 1 TO ERROR-SW
         MOVE NO-ELEMENT-MSG TO CDSIMSG
         MOVE Q-EL TO CDSIELNM
         MODIFY MAP CDSIMAP1 TEMPORARY
             FOR CDSIELNM ATTRIBUTES BRIGHT
         GO TO 4999-EXIT.
     MOVE CDSIELNM TO CIO-I-TABLE.
 4999-EXIT.
     EXIT.
*************************************************************
*                                                           *
* ROUTINE - 5000-CALL-COMMAND FACILITY                      *
*                                                           *
* THIS ROUTINE CALLS COMMAND FACILITY,                      *
* PASSING THE SEVEN PARAMETERS THAT                         *
* ARE REQUIRED.  IF THE RETURN CODE FROM                    *
* COMMAND FACILITY IS GOOD (ALL                             *
* BINARY ZEROS) THE FIRST TEN LINES FROM THE CIOF OUTPUT    *
* WORKAREA (THE COMMAND FACILITY SYSLST FILE)               *
* ARE MOVED TO THE MAP.                                     *
* IF THE RETURN CODE FROM                                   *
* COMMAND FACILITY IS BAD (NOT BINARY ZEROS) AN             *
* ERROR MESSAGE IS DISPLAYED WITH THE ERROR CODE.           *
*                                                           *
*************************************************************
 5000-CALL-COMMAND FACILITY.
     TRANSFER CONTROL TO 'IDMSOCF' RETURN
         USING CIO-PARM1
               CIO-PARM2
               CIO-PARM3
               CIO-PARM4
               CIO-PARM5
               CIO-PARM6
               CIO-PARM7.
     IF CIO-IO-RC NOT = GOOD-RC
         MOVE CIO-IO-RC TO CIO-ERROR-CODE
         MOVE CIO-ERROR-MSG TO CDSIMSG
         GO TO 5999-EXIT.
     PERFORM 5100-MOVE-COMMAND FACILITY-OUTPUT THRU 5109-EXIT
         VARYING SUB FROM 1 BY 1
             UNTIL COMMAND-FACILITY-EOF.
     MOVE OK-MSG         TO CDSIMSG.
     GO TO 5999-EXIT.
```

```
 5100-MOVE-COMMAND FACILITY-OUTPUT.
     MOVE CIOF-OUTPUT-LINE(SUB) TO CDSILINE(SUB).
     IF (SUB = 10) OR (SUB = CIOF-O-SIZE-US)
         MOVE 1 TO COMMAND-FACILITY-EOF-SW.
 5109-EXIT.
     EXIT.
 5999-EXIT.
     EXIT.

     EJECT
*************************************************************
*                                                           *
* ROUTINE - 8000-INITILIZE-MAP                              *
*                                                           *
* THIS ROUTINE DOES THE IDMS MAP BINDS.                     *
*                                                           *
*************************************************************
 8000-INITILIZE-MAP.
     COPY IDMS MAP-BINDS.
 8099-EXIT.
     EXIT.
 EJECT
     COPY IDMS IDMS-STATUS.
 IDMS-ABORT.
 IDMS-ABORT-EXIT.
     EXIT.
```

Index

A

APPLY command 3-15, 3-18
  associating the ENTER key with 3-20

B

batch processing
  See IDMSBCF

C

canceling edits and redisplaying 3-23
CIO block A-4
CIOF block A-4
CLEAR command 3-15, 3-18
  description 3-18
coding considerations 1-6
command area
  online OCF 3-5
command facility compiler
  subprogram, as a A-3, A-16
Command facility entity types, identifying 1-7
Command facility keyword 1-7
command facility statements, coding for 1-7
commands 3-15—3-27
  table of 3-15
comments, coding 1-6
compiler session control 1-6
components
  batch 1-4
  online 1-4
CONNECT statement 1-6, 1-8
connecting to a dictionary 1-6

D

description 1-4
dictionaries, accessing 1-7
dictionary, connecting to a 1-6, 1-8
DISPLAY LINE command 3-15, 3-18
DISPLAY PAGE
  description 3-19
DISPLAY PAGE command 3-15, 3-19

E

EDIT command 3-14
editing and saving
  OCF-language modules 3-14
editing commands 3-26—3-27
END command 3-15, 3-20
ENDIDD statement 1-7
ENTER command 3-15, 3-20
error messages
  OCF, online 3-7
escape character 3-26
ESCAPE command 3-15, 3-21
EXECIDD statement 1-7
Executing statements
  See also APPLY
  See also UPDATE
  as part of a command list 3-12
  at the system prompt 3-11
  at the system prompt using OCFX 3-10

F

FIND command 3-15, 3-21
formatting commands 1-6
formatting, SET OPTIONS 1-9

H

HELP command 3-15, 3-17, 3-21

I

IDMSBCF
  JCL 2-4—2-18
  job stream 2-4
  messages 2-4
  output 2-4
  overview 2-3
  return codes 2-4
  statement processing 2-4
IDMSBCF (batch processing) 2-3—2-18
input statements 1-4
INSERT command 3-16, 3-22
  used with APPLY 3-18

J

JCL for IDMSBCF 2-4—2-18
job control language
  See JCL

L

local mode
  IDMSBCF (BS2000/OSD) 2-17

O

OCF
  conducting a session 3-4—3-8
  editing commands 3-26—3-27
  session-control (top-line) commands 3-15—3-25
OCF (online processing) 3-3—3-27
OCF-language modules, editing and saving 3-14
OCF, online 3-7
  error messages 3-7
OCFX 3-10—3-13
  status codes 3-12
online processing
  See OCF
operating system commands
  See JCL
overview 1-3—1-16

P

parameters for SET OPTIONS 1-10—1-13
PRINT command 3-16, 3-22

R

REPEAT command 3-16, 3-23
repeating a line in a work file 3-23
RESHOW command 3-16, 3-23
restoring the screen and work file 3-24

S

SAVE command 3-14
searching, for a string 3-21
session
  online OCF 3-5
session control, SET OPTIONS 1-9
session-control commands
  APPLY 3-18
  CLEAR 3-18
  DISPLAY LINE 3-18
  DISPLAY PAGE 3-19
  END 3-20
  ENTER 3-20
  entering 3-15
  ESCAPE 3-21
  FIND 3-21
session-control commands *(continued)*
  HELP 3-21
  INSERT 3-22
  PF keys for 3-17—3-18
  PRINT 3-22
  REPEAT 3-23
  RESHOW 3-23
  SUSPEND 3-23
  SWAP 3-24
  SWITCH 3-24
  UPDATE 3-25
SET OPTIONS statement 1-9
SET OPTIONS statement, parameters 1-10—1-13
SET OPTIONS, examples 1-14
SET OPTIONS, terminating 1-13
statements you can input 1-4
statements, executing 3-8
string location 3-21
subroutine A-3
SUSPEND command 3-16, 3-23
SWAP command 3-16, 3-24
SWITCH command 3-16, 3-24

T

terminating an OCF session
  END command 3-20
  SUSPEND command 3-23
Transfer Control Facility (TCF) 3-24
transferring between development tools using
  SWITCH 3-24

U

UPDATE command 3-16, 3-25
  associating the ENTER key with 3-20