CA IDMS™

Common Facilities Guide

Release 18.5.00, 2nd Edition

ca technologies

CA Technologies Product References

This document references the following CA products:

- CA ADS™
- CA ADS™ Batch
- CA IDMS™/DB
- CA IDMS™/DC
- CA IDMS™/DC or CA IDMS™ UCF (DC/UCF)
- CA IDMS™ UCF (UCF)
- CA OLQ™ Online Query for CA IDMS™ (CA OLQ)
- CA Optimizer®/II
- CA SymDump® Batch

Contact CA Technologies

Contact CA Support

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

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

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

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

Documentation Changes

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

- BUFFERSTAT Report Field Descriptions (see page 166)—Added section.
- z/VSE File Parameters (see page 169)—Added section.
- Parameter Descriptions (see page 146), Parameter Summary (see page 141)—The WORK parameter was added.
- Parameter Summary (see page 141)—Added the z/VSE File-Related Parameters table.
- Parameter Descriptions (see page 146)—The description of the SAVE_SQL_SYNTAX parameter was updated.

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

- Parameter Summary (see page 141)—The PRMPT_MSG and SORT FIELD MAX LEN parameters were added to the Miscellaneous Runtime Directives table. Added the DBIO_HICCUP parameter to the Performance-Related Parameters table.
- Parameter Descriptions (see page 146)—The description of the PRMPT_MSG, SORT FIELD MAX LEN, and DBIO_HICCUP parameters were added.

Contents

Chapter 1: Introduction

This guide provides information on how to use the following CA IDMS/DB Database facilities and the SYSIDMS parameter file:

- The Command Facility lets you submit command statements in a batch or online environment.
- The Online Compiler Text Editor lets you edit compiler output and resubmit it as input using the CA IDMS development tools.
- The Transfer Control Facility lets you transfer between CA IDMS development tools.
- The SYSIDMS parameter file contains parameters that you can add to a batch job running in local mode or under the central version. These parameters let you specify environment requirements, runtime directives, and operating system-dependent information.

This section contains the following topics:

Syntax Diagram Conventions (see page 9)

Syntax Diagram Conventions

The syntax diagrams presented in this guide use the following notation conventions:

`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.

`italicized lowercase`

Represents a value that you supply.

`lowercase bold`

Represents a portion of the syntax shown in greater detail at the end of the syntax or elsewhere in the document.

◄─

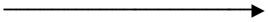
Points to the default in a list of choices.

►►──────────────────

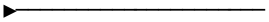
Indicates the beginning of a complete piece of syntax.

──────────────────►◄

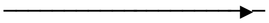
Indicates the end of a complete piece of syntax.



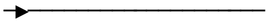
Indicates that the syntax continues on the next line.



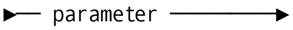
Indicates that the syntax continues on this line.



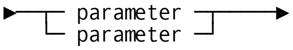
Indicates that the parameter continues on the next line.



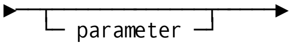
Indicates that a parameter continues on this line.



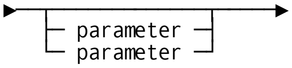
Indicates a required parameter.



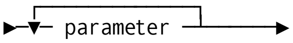
Indicates a choice of required parameters. You must select one.



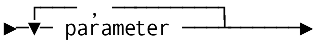
Indicates an optional parameter.



Indicates a choice of optional parameters. Select one or none.



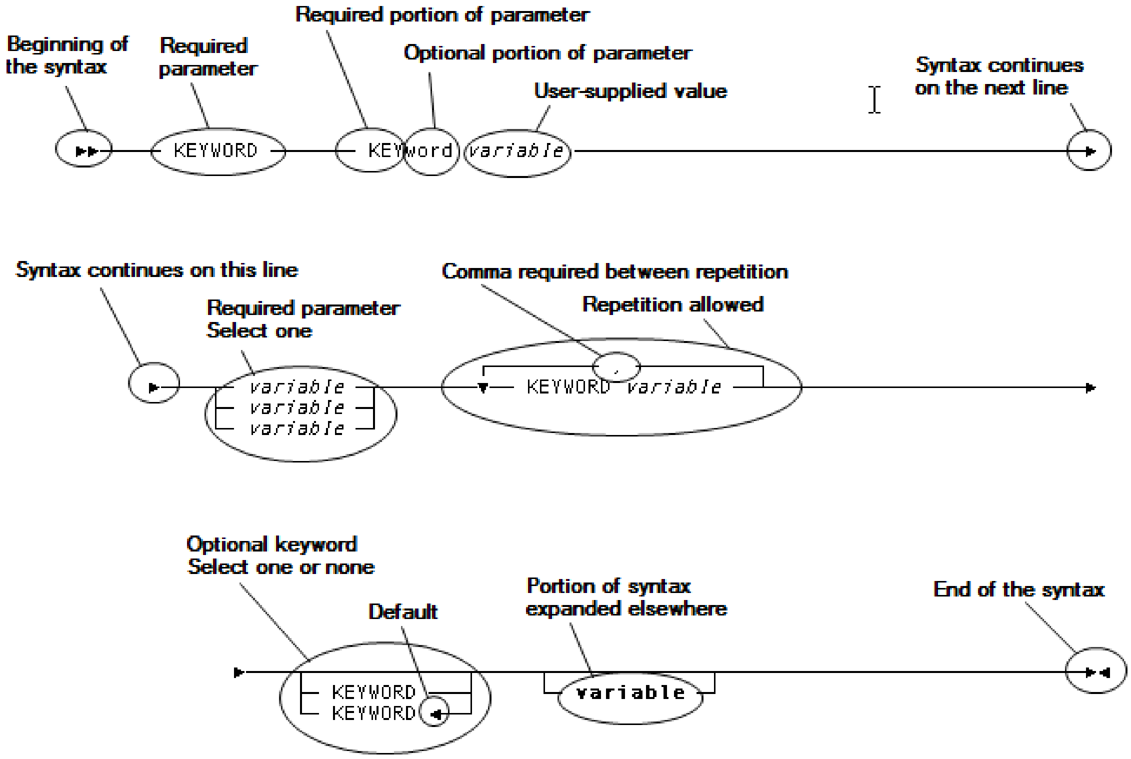
Indicates that you can repeat the parameter or specify more than one parameter.



Indicates that you must enter a comma between repetitions of the parameter.

Sample Syntax Diagram

The following sample explains how the notation conventions are used:


Beginning of the syntax
Required parameter
Required portion of parameter
Optional portion of parameter
User-supplied value
Syntax continues on the next line
KEYWORD
KEYword
variable
Syntax continues on this line
Required parameter Select one
Comma required between repetition
Repetition allowed
variable
variable
variable
,
KEYWORD variable
Optional keyword Select one or none
Default
Portion of syntax expanded elsewhere
End of the syntax
KEYWORD
KEYWORD
variable

Chapter 2: 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.

Types of Statements You Can Submit

This section describes the types of statements you can submit using the command facility.

Physical DDL Statements: Use physical DDL statements for SQL and non-SQL defined databases to define the following:

- Database name tables
- DMCLs
- Segments

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

- CALC keys
- Functions
- Indexes
- Procedures
- Referential constraints
- Schemas
- Table Procedures
- Tables
- Views

Coding Considerations

Follow the coding guidelines described in the following table when you use the command facility.

Guideline for:	Description
Input	For batch, the input must be in columns 1 to 72 or 80. The default is 72. You can change the default by using the SYSIDMS parameter BCF_INPUT_80. For more information, see SYSIDMS parameter file chapter. For online, the input must be in columns 1 to 79. Can be 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 the *CA IDMS Database Administration Guide*. ■ Logical DDL (SQL-defined databases)—See the *CA IDMS SQL Reference Guide*. ■ SQL DML—See the *CA IDMS SQL Reference Guide*. ■ Utilities—See the *CA IDMS Utilities Guide*. ■ Security administration—See the *CA IDMS Security Administration Guide*.
Statement terminator	End each statement with a command terminator or delimiter. Initially the command terminator is the semicolon (;), but it can be modified by the SET OPTIONS COMMAND DELIMITER command. The command delimiter 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.

Guideline for:	Description
Connecting to a dictionary	Use a CONNECT statement in a batch or online command facility session (see Connecting to a Dictionary)
Session control	Use the SET OPTIONS statement (see Using SET OPTIONS to Select Options)
Output formatting control	Use the SET OPTIONS statement (see Using SET OPTIONS to Select Options)

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 singly or as a block of statements. These two methods are described as follows. Also the considerations are described 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.

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.

  Note: For more information about SYSIDMS parameters, see SYSIDMS Parameter File.

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

More Information

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

Using SET OPTIONS to Select Options

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

- Control the processing of statements
- Format the output from the execution of SQL SELECT statements
- Control input and output of the batch command facility (IDMSBCF)
- Change the command delimiter

Syntax

```
                      ┌───────────────────────────────┐
►►─── SET OPTIONS ───▼┬── session-option ──────────┬┴──────────cmd-delimiter──►◄
                      ├── format-option ───────────┤
                      ├── io-option ───────────────┤
                      └── cmd-delimiter-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 ◄─┘
```

Expansion of io-option

```
►►─┬─ OUTPUT TO ─────┬─ SYSLST ◄────────┬──────────────────────┬────►◄
   │                 └─ output-ddname ──┘                      │
   └─ INPUT FROM ─────┬─ SYSIPT ◄───────┬──────────────────────┘
                      └─ input-ddname ──┘
```

Expansion of cmd-delimiter-option

```
►►──── COMMAND DELIMITER ────┬─ DEFAULT ◄───────────────┬────────────────►◄
                             └─ 'new-cmd-delimiter' ────┘
```

Parameters

cmd-delimiter

The current command delimiter to terminate the SET OPTIONS command. Initially the command delimiter is the semicolon(;) character, but it can be modified with the SET OPTIONS **cmd-delimiter-option** command

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.

AUTOCOMMIT

Specifies AUTOCOMMIT options after successful statement execution (for more information about options for statements that return errors, see the ON ERROR parameter). 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.		

Option	IDMSBCF	OCF	OCFX
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.

io-option

On a SET OPTIONS statement, specifies options for controlling the input and output of the command facility. The io-option is for IDMSBCF (batch) only.

OUTPUT TO

Specifies where to write data output.

SYSLST

Writes data output to SYSLST. If the prior assignment of the OUTPUT stream was not SYSLST, the prior OUTPUT assignment is closed.

output-ddname

Specifies the z/OS DD name, z/VSE file name link name of a sequential data set to use for writing the data output.

When output is assigned to *output-ddname*, these rules apply:

- WIDTH PAGE is automatically set to the record length (or maximum record length for variable record files) that was specified when the file associated with *output-ddname* was created. If no record length and record format were specified, the record format defaults to variable and the record length to block size - 4; if no block size was specified, a block size of 4096 is used.
- HEADINGS are set to OFF.
- The "non-data" information like the echoed command, eventual headers, the number of rows processed, and the SQL return code are output to SYSLST.
- Output data are not prefixed by "*+".
- Data are represented in string format, not in the native format. For example, a column defined as INT with value 12345678 is internally stored as a 4-byte binary value X'00BC614E'; in the output data however, the column value is 8-byte character string '12345678'.
- The width of each column in the output file is determined by the larger value of the column width and the column header. For example, a column named "Date", defined as CHAR(10) uses 10 positions in the output file; a column named "MiddleInitial", defined as CHAR(1) uses 13 positions.
- IDMSBCF inserts two blanks in between successive columns.
- The *output-ddname* file is closed on the next SET OPTIONS OUTPUT or at program end.

INPUT FROM

Specifies where to read input.

SYSIPT

Reads input from SYSIPT.

input-ddname

Specifies the DD name of a sequential data set to use for reading commands.

When input is assigned to *input-ddname*, these rules apply:

- Input from *input-ddname* can be any type and length supported by the operating system, that is, input is not limited to 80 character lines.
- Columns 73 through 80 of the input are NOT considered as a line sequence number, that is, they should contain valid input data.
- End-of-file on the *input-ddname* file automatically reassigns input to SYSIPT.

cmd-delimiter-option

On a SET OPTIONS statement, specifies the command facility terminator.

COMMAND DELIMITER

Specifies the character string whose value will be used to delimit command facility statements after the SET OPTIONS command has been executed.

new-cmd-delimiter

Specifies the character string literal to be used as a delimiter. '*Delimiter*' must be a 1- to 32-character string.

DEFAULT

Specifies that the default of a semicolon (;) will be used as a delimiter.

Usage

Statement terminator

Use the command delimiter, by default a semicolon, to terminate a SET OPTIONS statement. The use of an alternate command delimiter is required when entering multi-statement SQL routine bodies using the CREATE PROCEDURE or CREATE FUNCTION SQL DDL commands. According to the SQL procedural language, multiple SQL statements must be separated by the semicolon. However, using the semicolon also as the command terminator would truncate the CREATE command after the first semicolon, and any statements thereafter would erroneously be interpreted as new commands for the command facility and not as statements that make up the rest of the SQL routine body.

Note: Specifying a command terminator string replaces the previous specified one or the default, the semicolon, if none was specified. The specification of a command delimiter, just as any SET OPTIONS parameter remains in effect until a new SET OPTIONS COMMAND DELIMITER is issued or until the end of the command facility session.

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, formatting, io and command delimiter 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.

Input and output assignment

You can use the OUTPUT TO parameter to output the resulting data of, for example, SQL commands to an intermediate file, which can then be used as input to IDMSBCF or a user written program.

Combining the OUTPUT TO and INPUT FROM parameters allow you to write IDMSBCF scripts to perform the following tasks:

- Unload/Load or copy of selective table(s) using SQL DML.
- Automatic access module recompile script for all access modules that are affected by an update statistics or change in table or any other condition that can be detected by looking in the catalog/dictionary.
- Build LOAD file for loading data using SQL DML.
- Build XML scripts to unload/load data from/to CA IDMS to/from XML documents.

Examples

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.

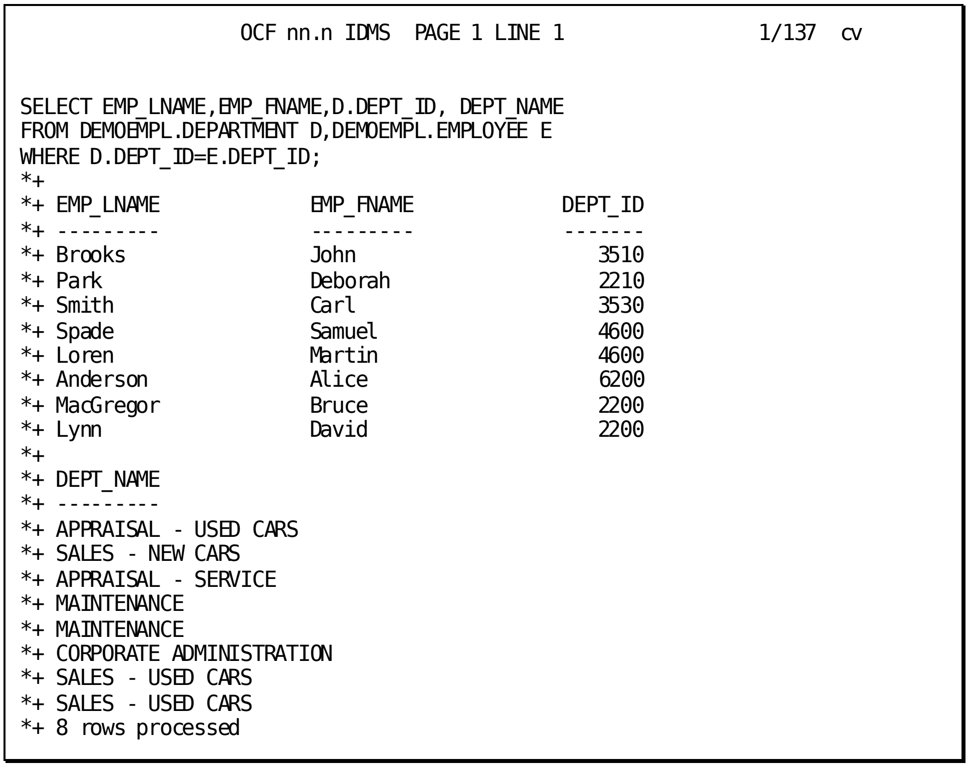
```
set options title 'employee list'
        headings on
        underline *
        lines 24
        width page 80
        width char 10
        width num 5
        width column 2 6
        pagebreak 2 4
        column wrap off;
```

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 is not 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.



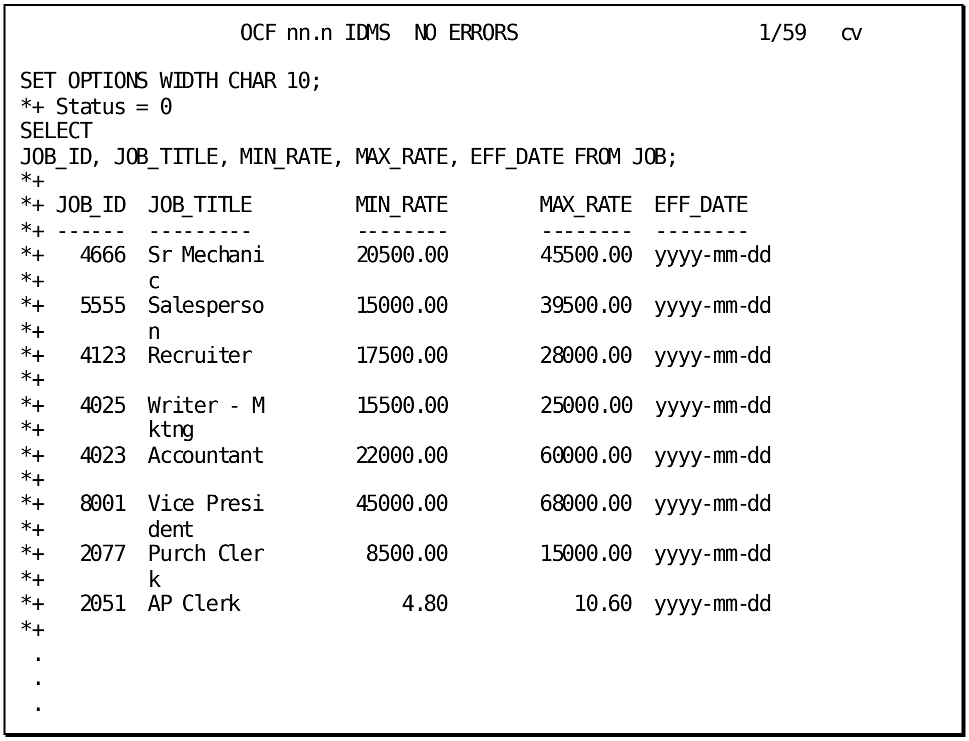
```
                  OCF nn.n IDMS  PAGE 1 LINE 1              1/137  cv


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 nn.n IDMS  NO ERRORS                    1/59   cv

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  yyyy-mm-dd
*+         c
*+   5555  Salesperso      15000.00        39500.00  yyyy-mm-dd
*+         n
*+   4123  Recruiter       17500.00        28000.00  yyyy-mm-dd
*+
*+   4025  Writer - M      15500.00        25000.00  yyyy-mm-dd
*+         ktng
*+   4023  Accountant      22000.00        60000.00  yyyy-mm-dd
*+
*+   8001  Vice Presi      45000.00        68000.00  yyyy-mm-dd
*+         dent
*+   2077  Purch Cler       8500.00        15000.00  yyyy-mm-dd
*+         k
*+   2051  AP Clerk            4.80           10.60  yyyy-mm-dd
*+
 .
 .
 .
```

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.

Sample IDMSBCF script

The following IDMSBCF example is a fairly generic script to unload/load or copy a table or set of tables. The sample script allows null values; however, it does not allow data containing quotes, more exotic data types, such as GRAPHIC, VARGRAPHIC, BINARY, and so on.

Input Script

```
-------------------------------------------------------------------------
-- This scripts copies the rows from a source table to a target table.
-- It is assumed that the target table is already defined
-------------------------------------------------------------------------
--
-- Helper view to set the params of the Table copy
--
drop   view usera01.CopyTabParm;
create view usera01.CopyTabParm as
  select SCHEMA     as SrcSchema
       , Name       as SrcTable
       , 'USERA01'  as TgtSchema  -- Set value of TgtSchema
       , 'EMPLOYEE' as TgtTable   -- Set value of TgtTable
   from SYSTEM.TABLE
  where SCHEMA = 'DEMOEMPL'       -- Set value of SrcSchema
    and NAME   = 'EMPLOYEE'       -- Set value of SrcTable
;
```

```
--
-- Create the Unload syntax
--
set options OUTPUT to Unload;

select 'select ''insert into '
     || trim(TgtSchema) || '.' || trim(TgtTable)
     || ' VALUES( '''
     , '-'||'-', 0 as sequence
from usera01.CopyTabParm
union
SELECT
 '||''' || SUBSTR(', ', CAST(1/NUMBER as SMALLINT) + 1, 1)
 || '''||TRIM(VALUE('
 || SUBSTR('CAST(      ''''''''||     ''''''''||CAST('
         , (11 * (  LOCATE(TYPE, 'CHARACTER           ', 1)
                  + LOCATE(TYPE, 'VARCHAR             ', 1)
                  + LOCATE(TYPE, ' DATE               ', 1)) + 1)
         , 11)
 || trim(NAME) || ' '
 || SUBSTR(
    'as char(10))      ||''''''''            as char(10))||'''''''''
         , (18* (  LOCATE(TYPE, 'CHARACTER           ', 1)
                 + LOCATE(TYPE, 'VARCHAR             ', 1)
                 + LOCATE(TYPE, ' DATE               ', 1)) + 1)
         ,18)
 || ',''NULL''))'
  , '-'||'-', NUMBER as sequence
FROM SYSTEM.COLUMN, usera01.CopyTabParm
 WHERE TABLE  = SrcTable
   and schema = SrcSchema
union
select '||'');'' from '
     || trim(SrcSchema) || '.' || trim(SrcTable) || ';'
     , '-'||'-', 99999 as sequence
from usera01.CopyTabParm
order by sequence
;
--
-- Create the Load syntax for the new Table
--
set options OUTPUT to Load;
set options INPUT from Unload;

--
-- Load the new Table
--
set options OUTPUT to SYSLST;
set options INPUT from Load;
```

Output from Sample Generic Table Copy Script

Unload OUTPUT

```
select 'insert into USERA01.EMPLOYEE VALUES( ' -- 0
||' '||TRIM(VALUE(CAST(      EMP_ID as char(10))      ,'NULL'))          --  1
||','||TRIM(VALUE(CAST(      MANAGER_ID as char(10))      ,'NULL'))      --  2
||','||TRIM(VALUE(''''||     EMP_FNAME ||''''            ,'NULL'))       --  3
||','||TRIM(VALUE(''''||     EMP_LNAME ||''''            ,'NULL'))       --  4
||','||TRIM(VALUE(CAST(      DEPT_ID as char(10))      ,'NULL'))         --  5
||','||TRIM(VALUE(''''||     STREET ||''''            ,'NULL'))          --  6
||','||TRIM(VALUE(''''||     CITY ||''''            ,'NULL'))            --  7
||','||TRIM(VALUE(''''||     STATE ||''''            ,'NULL'))           --  8
||','||TRIM(VALUE(''''||     ZIP_CODE ||''''            ,'NULL'))        --  9
||','||TRIM(VALUE(''''||     PHONE ||''''            ,'NULL'))           --  10
||','||TRIM(VALUE(''''||     STATUS ||''''            ,'NULL'))          --  11
||','||TRIM(VALUE(CAST(      SS_NUMBER as char(10))      ,'NULL'))       --  12
||','||TRIM(VALUE(''''||CAST(START_DATE as char(10))||'''','NULL'))      --  13
||','||TRIM(VALUE(''''||CAST(TERMINATION_DATE as char(10))||'''','NULL')) --  14
||','||TRIM(VALUE(''''||CAST(BIRTH_DATE as char(10))||'''','NULL'))      --  15
||');' from DEMOEMPL.EMPLOYEE; -- 99999
```

Load OUTPUT

```
insert into USERA01.EMPLOYEE VALUES(  2299,NULL,'Samuel
','Spade               ',4600,'47 London St
','Canton              ','MA','02020
',NULL,'L',33892200,'1991-02-04',NULL,'1958-01-09');
insert into USERA01.EMPLOYEE VALUES(  3411,2894,'Catherine
','Williams            ',5200,'566 Lincoln St
','Boston              ','MA','02010
',NULL,'A',83356561,'1993-09-30',NULL,'1967-10-28');
insert into USERA01.EMPLOYEE VALUES(  4773,3082,'Janice
','Dexter              ',3510,'399 Pine St
','Medford             ','MA','02432
','5083847566','A',89675632,'1997-06-14',NULL,'1969-11-19');

   ...
   ...
   ...

insert into USERA01.EMPLOYEE VALUES(  3118,3222,'Alan
','Wooding             ',4500,'196 School St
','Canton              ','MA','02020
','5083766984','A',98746783,'1992-11-18',NULL,'1969-05-17');
insert into USERA01.EMPLOYEE VALUES(  3769,2894,'Julie
','Donelson            ',3520,'14 Atwood Rd
','Grover              ','MA','02976
','5084850432','A',67783532,'1994-08-31',NULL,'1967-08-15');
```

Using an alternate command delimiter

The definition of the following SQL procedure requires the use of an alternate command delimiter to avoid a collision of the default, the semicolon, with the delimiter in the SQL procedural language.

```
set options command delimiter '++';
drop procedure PRODUCTION.PROCESS ++
commit++
create procedure PRODUCTION.PROCESS
 (PROC_TYPE integer,PROC_VALUE char(2_))
   external name DPROCESS language SQL
begin
  set PROC_TYPE = 12;
  set PROC_VALUE = 'High';
end
++
set options command delimiter DEFAULT ++
```

Chapter 3: Batch Processing

The Batch Command Facility (IDMSBCF) is the batch component of the command facility.

This section contains the following topics:

Batch Processing Overview

To submit statements in a batch environment, you enter the statements in an IDMSBCF job stream.

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.

Note: For more information about the SET OPTIONS statement, see Using SET OPTIONS to Select Options.

IDMSBCF Job Stream

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

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

Batch Output

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

- 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 the following:

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

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.

JCL for z/OS

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

IDMSBCF (z/OS)

```
//stepname EXEC PGM=IDMSBCF,REGION=1024K
//STEPLIB  DD   DSN=idms.dba.loadlib,DISP=SHR
//         DD   DSN=idms.custom.loadlib,DISP=SHR
           DD   DSN=idms.cagjload,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)
/*
```

Note: For more information about SYSIDMS parameters, see SYSIDMS Parameter File.

Stepname	Name of the job step
idms.dba.loadlib	Data set name of the load library containing the DMCL and database name table load modules
idms.custom.loadlib	Data set name of the load library containing customized CA IDMS executable load modules
idms.cagjload	Data set name of the load library containing the CA IDMS executable modules that do not require customization
sysctl	Ddname of the SYSCTL file
idms.sysctl	Data set name of the SYSCTL file (the link to the central version)
dcmsg	Ddname of the system message (DDLDCMSG) area
idms.sysmsg.ddldcmsg	Data set 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	Data set name of the system dictionary catalog (DDLCAT) area
dccatl	Ddname of the system dictionary catalog load (DDLCATLOD) area
idms.system.ddlcatlod	Data set name of the system dictionary catalog load (DDLCATLOD) area
dccatx	Ddname of the system dictionary catalog index (DDLCATX) area
idms.system.ddlcatx	Data set 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	Data set name of the SQL catalog area
sqllod	Ddname of the SQL catalog load area
idms.syssql.ddlcatlod	Data set name of the SQL catalog load area
sqlxdd	Ddname of the SQL catalog index area
idms.syssql.ddlcatx	Data set name of the SQL catalog index area
userdb	Ddname of the user database file
user.userdb	Data set name of the user database file

If You Submit Utility Statements

For more information about the file assignments specific to utility statements, see the *CA IDMS Utilities Guide*.

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	Data set name of the local mode scratch (DDLOCSCR) area
secdd	Ddname of the system user catalog (DDLSEC) area
idms.sysuser.ddlsec	Data set name of the system user catalog (DDLSEC) area

dictdb	Ddname of the application dictionary definition (DDLDML) area
idms.appldict.ddldml	Data set name of the application dictionary definition (DDLDML) area
dloddb	Ddname of the application dictionary definition load (DDLDCLOD) area
idms.appldict.ddldclod	Data set name of the application dictionary definition load (DDLDCLOD) area
sysjrnl	Ddname of the tape journal file
idms.tapejrnl	Data set 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*

Commands for z/VM

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

IDMSBCF (z/VM)

```
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.

Note: When executing local mode or batch-to-CV mode execs in CMS, filedefs for CDMSLIB loadlibs are optional. However, if omitted, and multiple EXECOS OSRUN statements are coded in the same EXEC, you must code the following CMS statement prior to the first EXECOS OSRUN statement in the EXEC:

```
SET STORECLR ENDCMD
```

You must also code the following CMS statement following the last EXECOS OSRUN statement:

```
SET STORECLR ENDSVC
```

Runtime Parameters

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

Note: For more information about SYSIDMS parameters, see SYSIDMS Parameter File.

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 z/VM commands:

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

JCL for z/VSE

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

IDMSBCF (z/VSE)

```
// 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 data set output from IDMSBCF
temp.bcf	File ID of the data set 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.

Note: For more information about SYSIDMS parameters, see SYSIDMS Parameter File (see page 139).

Executing in Local Mode

To execute IDMSBCF 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 media) to reflect the filenames and definitions in use at your site. Reference IDMSLBLS as shown in the previous z/VSE JCL job stream.

```
* -------- LIBDEFS --------
// LIBDEF  *,SEARCH=idmslib.sublib
// LIBDEF  *,CATALOG=user.sublib
/*  ------------------------- LABELS -------------------------
// DLBL    idmslib,'idms.library',2099/365idms
// 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

Chapter 4: Online Processing

The Online Command Facility (OCF) is the online component of the command facility. It allows you to submit command statements interactively and see the resulting output on a display screen.

You can execute and submit statements in the OCF environment in one of the following ways:

- Dynamically
- At the system prompt
- As part of a system command list (CLIST)

This section contains the following topics:

Beginning an OCF Session

To begin an online session, you need to do the following:

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:

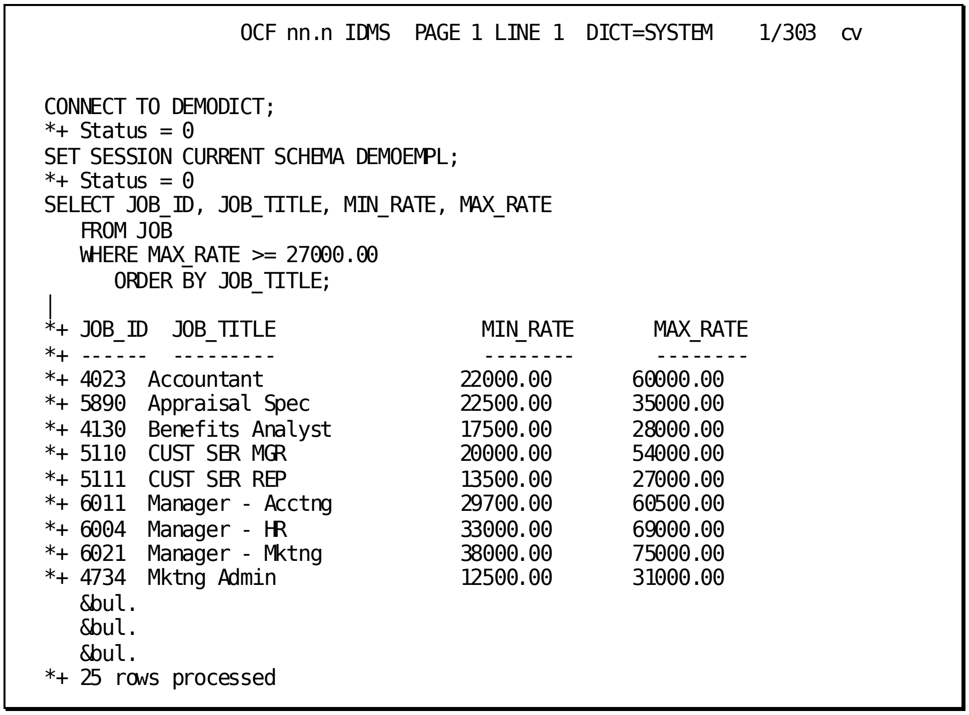
   ```
   ENTER NEXT TASK CODE: OCF Enter
   ```

3. Optionally, enter a CONNECT statement to establish a dictionary to which to connect during the session (see Connecting to a Dictionary).
4. Optionally, enter a SET OPTIONS statement to establish session-specific processing and output formatting options (see Using SET OPTIONS to Select Options).

OCF Screen Format

OCF uses a standard screen, which contains the following:

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



```
                  OCF nn.n IDMS  PAGE 1 LINE 1  DICT=SYSTEM    1/303  cv


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
   &bul.
   &bul.
   &bul.
*+ 25 rows processed
```

Top Line

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

- *Command area*—Provides 20 spaces in columns 2 through 21 for entering commands that manipulate the work file or communicate with OCF; these commands are described in Using Top-line Commands.
- *Name area*—Displays the OCF header and the release number.
- *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.
- *Dictionary area*—Displays the current dictionary if you used the system DCUF SET DICTNAME statement or a CONNECT TO statement.
- *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

In the area that follows 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).

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 statements, utility statements, 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 Executing Statements Dynamically).
- Save the work file as a dictionary module (see Creating, Editing, and Saving Modules).
- Display the contents of any dictionary module for editing (see Creating, Editing, and Saving Modules).

Session Options

You can tailor your processing and output options for the session using the SET OPTIONS statement (see Using SET OPTIONS to Select Options). 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 Using Top-line Commands) or use the program function (PF) key assigned to the desired command function (see Using Control Keys (see page 99)).

Editing Options

You can use line-editing commands to edit the statements contained in your OCF work file (see Using Line Commands).

Coding Considerations

For more information about coding considerations, see Coding Considerations.

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 re-execute 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.

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 does the following:

  - 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.

Executing Statements Dynamically

Enter statements in your work file according to the coding considerations described in Coding Considerations. 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.

Executing Statements Stored in Modules (OCFX)

You can use the OCFX statement to execute statements contained in modules with the language OCF. The following are the 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 that lists and describes OCFX status codes.

Executing at the System Prompt

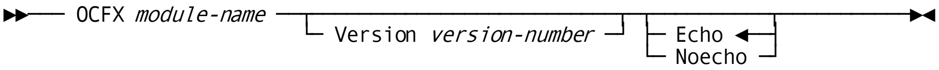
To execute OCF-language modules at the system prompt, you enter the OCFX statement after the prompt.

Example:

ENTER NEXT TASK CODE: **OCFX ddlmod01 version 3** Enter

Syntax

OCFX statement



Parameters

OCFX

Executes the statements in a 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.

Note: For more information about the LANGUAGE parameter of the MODULE statement, see 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 Using SET OPTIONS to Select Options) 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.

Executing as Part of a Command List

To execute OCF-language modules as part of a system command list (CLIST), you need to do the following:

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

Note: For more information about CLIST processing, see the *CA IDMS System Tasks and Operator Commands Guide*.

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.

Note: For more information about other status codes, see the *CA IDMS Messages and Codes Guide*.

Creating, Editing, and Saving Modules

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

Note: For more information about executing these modules, see Executing Statements Stored in Modules (OCFX).

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 Using Line Commands.

Note: If you do not specify a version, the default is the version number set up in the command facility.

Chapter 5: Using the Online Text Editor

The online compiler text editor is available in two types of CA IDMS development tools:

- Online compilers—Throughout the manual, the term *online compilers* refers to the following CA IDMS development tools:
  - DDDL compiler
  - Schema compiler
  - Subschema compiler
  - System generation compiler
- IDD Menu Facility—The online text editor provides two types of editing commands:
  - *Top-line commands* allow you to direct a development tool session. You enter top-line commands in the *command area* of the screen. Using Top-line Commands discusses these commands.
  - *Line commands* allow you to manipulate the contents of a work file. Line commands begin in the first column of a line. Using Line Commands discusses these commands.

The remainder of this chapter discusses screens available in the different development tools.

This section contains the following topics:

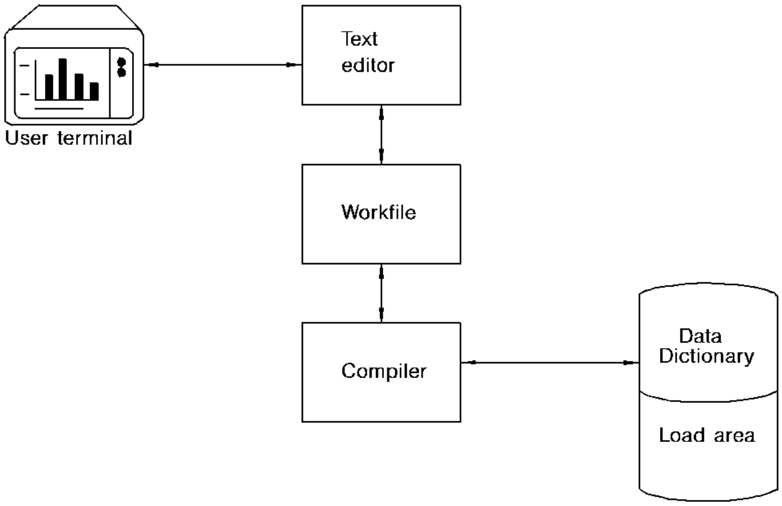
Online Compilers

Online compilers process requests to add, modify, replace, delete, or display entity-occurrence definitions. The online schema, subschema, DDDL, and system generation compilers use the same *text editor* to simplify the coding of compiler input.

The text editor operates independently of the compilers, writing input to and output from the compiler to a *work file* associated with each online session. A work file contains *pages* of compiler input or output. You use the online text editor to display and modify the contents of the work file so you can edit compiler output and resubmit it as input.

Online Compilation Flow

The following illustration shows the online compilation process:



Free-form Screen Format

The free-form screen for an online compiler session contains a preformatted pop line and an input/output (I/O) area in which you can enter commands and definitions in a free-form manner.

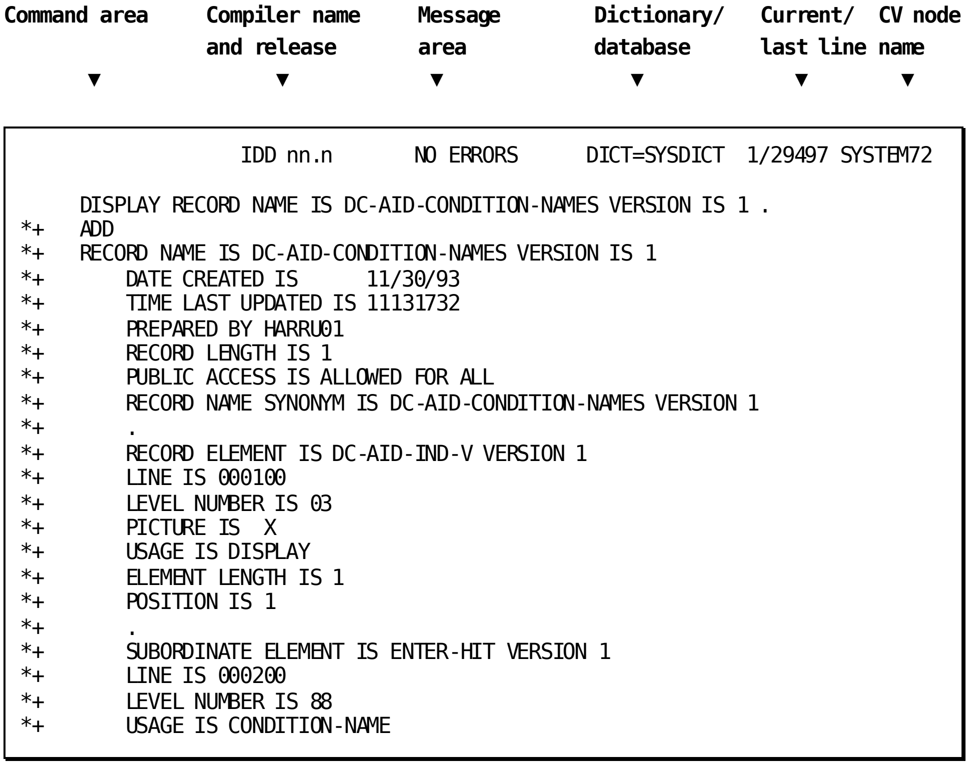
You can enter characters in columns one through 79 on free-form screens, except when you enter process modules for CA ADS Batch, in which case you should *not* enter characters past column 72.

The free-form screen consists of the following components:

- Command area—Comprises columns 2-21 for entering top-line commands
- Compiler name and release—Displays the online compiler name and release
- Message area—Displays one of the following:
  - Workfile page and line numbers
  - A NO ERRORS message
  - The number of error messages issued for the compile
  - The status if the PRINT command
- Dictionary/database names—Displays the name of the current dictionary and database, or the word BLK (if using block editing commands)
- Current/last lines—Displays the number of the current line and the last line of the work file, or the word EMPTY (if the work file is empty)
- CV node name—Displays the CV node name.

Example

The components of the free-form screen appear in the following example.



```
                    IDD nn.n       NO ERRORS      DICT=SYSDICT  1/29497 SYSTEM72

      DISPLAY RECORD NAME IS DC-AID-CONDITION-NAMES VERSION IS 1 .
*+    ADD
*+    RECORD NAME IS DC-AID-CONDITION-NAMES VERSION IS 1
*+        DATE CREATED IS      11/30/93
*+        TIME LAST UPDATED IS 11131732
*+        PREPARED BY HARRU01
*+        RECORD LENGTH IS 1
*+        PUBLIC ACCESS IS ALLOWED FOR ALL
*+        RECORD NAME SYNONYM IS DC-AID-CONDITION-NAMES VERSION 1
*+        .
*+        RECORD ELEMENT IS DC-AID-IND-V VERSION 1
*+        LINE IS 000100
*+        LEVEL NUMBER IS 03
*+        PICTURE IS  X
*+        USAGE IS DISPLAY
*+        ELEMENT LENGTH IS 1
*+        POSITION IS 1
*+        .
*+        SUBORDINATE ELEMENT IS ENTER-HIT VERSION 1
*+        LINE IS 000200
*+        LEVEL NUMBER IS 88
*+        USAGE IS CONDITION-NAME
```

The IDD Menu Facility

The *IDD Menu Facility* is a menu-driven mode of the online DDDL compiler. The Menu Facility provides an alternative to free-form input of many online DDDL commands.

Screen Formats

The IDD Menu Facility features three types of screen design:

- Fixed (non-pageable)
- Pageable
- Free-form

Each type of screen design is described as follows.

Using Fixed Screens

Fixed screens provide session, entity-occurrence, the program function (PF) key information. Fixed screens prompt you to enter the necessary specifications for the definitions you create.

Available Top-line Commands

You can use the following top-line commands on fixed screens:

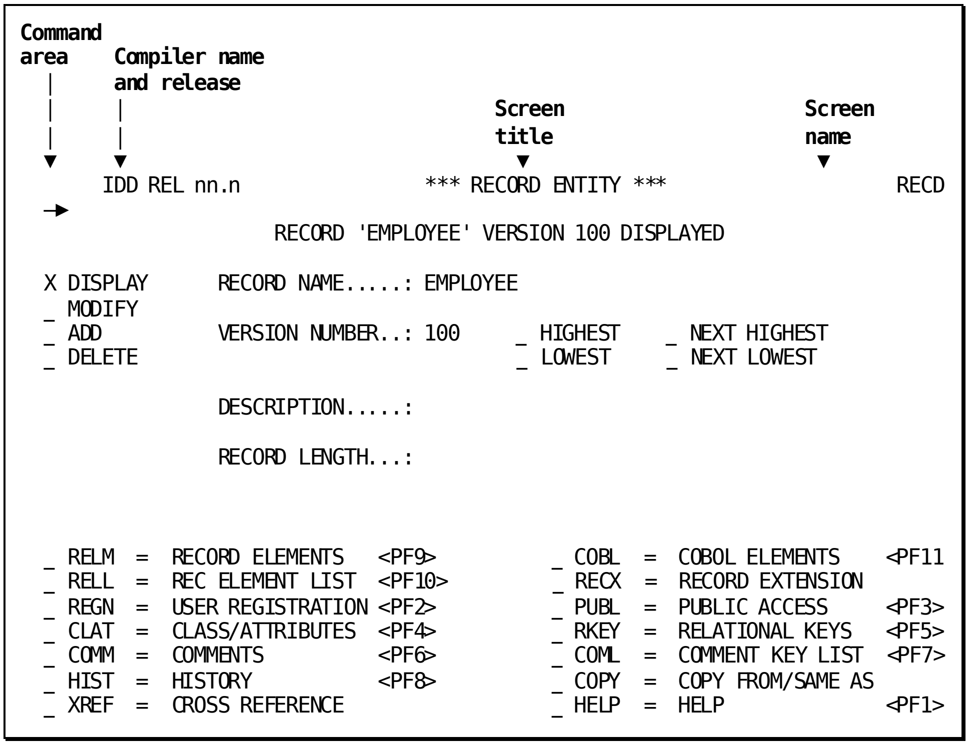
- HELP
- SUSPEND
- SWITCH

Note: For more information about top-line commands, see Using Top-line Commands.

Example

The following screen depicts that IDD Menu Facility fixed screens are divided into three areas:

- Heading and message area (first three lines)
- Specification area (beginning at X DISPLAY)
- Screen selection area (beginning at _RELM)



```
  IDD REL nn.n               *** RECORD ENTITY ***                  RECD
→
                     RECORD 'EMPLOYEE' VERSION 100 DISPLAYED

X DISPLAY      RECORD NAME.....: EMPLOYEE
_ MODIFY
_ ADD          VERSION NUMBER..: 100      _ HIGHEST     _ NEXT HIGHEST
_ DELETE                                  _ LOWEST      _ NEXT LOWEST

               DESCRIPTION.....:

               RECORD LENGTH...:


_ RELM  =  RECORD ELEMENTS    <PF9>         _ COBL  =  COBOL ELEMENTS    <PF11
_ RELL  =  REC ELEMENT LIST   <PF10>        _ RECX  =  RECORD EXTENSION
_ REGN  =  USER REGISTRATION  <PF2>         _ PUBL  =  PUBLIC ACCESS     <PF3>
_ CLAT  =  CLASS/ATTRIBUTES   <PF4>         _ RKEY  =  RELATIONAL KEYS   <PF5>
_ COMM  =  COMMENTS           <PF6>         _ COML  =  COMMENT KEY LIST  <PF7>
_ HIST  =  HISTORY            <PF8>         _ COPY  =  COPY FROM/SAME AS
_ XREF  =  CROSS REFERENCE                  _ HELP  =  HELP              <PF1>
```

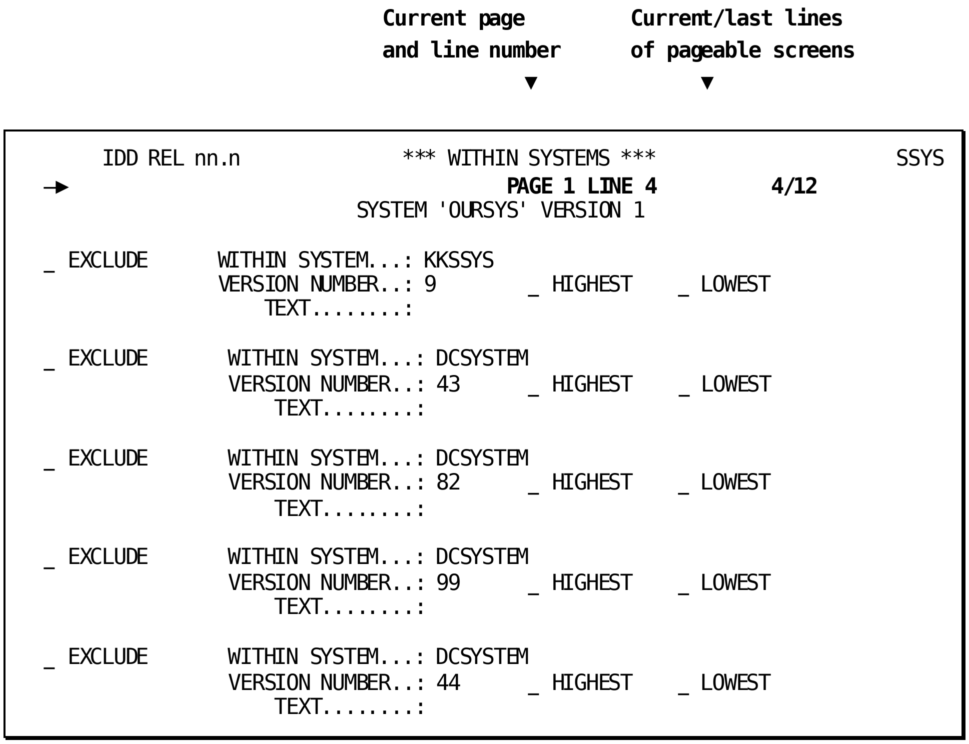
Using Pageable Screens

Pageable screens are similar to fixed screens in that both screens display a group of requested information (prompts). The difference is that pageable screens have multiple occurrences or multiple pages of the same group of prompts.

Two IDD Menu Facility pageable screens are the Systems Within Systems (SSYS) or Record Elements (RELM) screens. The DDDL compiler considers each *group* of lines that represents an entity to be one line of data.

Example

The following screen shows an example of the Systems Within Systems (SSYS). The DDDL compiler considers the lines beginning with _EXCLUDE and ending with TEXT... to be one line of data.



```
       IDD REL nn.n              *** WITHIN SYSTEMS ***                 SSYS
 →                                        PAGE 1 LINE 4         4/12
                               SYSTEM 'OURSYS' VERSION 1

 _ EXCLUDE      WITHIN SYSTEM...: KKSSYS
                VERSION NUMBER..: 9          _ HIGHEST    _ LOWEST
                    TEXT........:

 _ EXCLUDE       WITHIN SYSTEM...: DCSYSTEM
                 VERSION NUMBER..: 43        _ HIGHEST    _ LOWEST
                     TEXT........:

 _ EXCLUDE       WITHIN SYSTEM...: DCSYSTEM
                 VERSION NUMBER..: 82        _ HIGHEST    _ LOWEST
                     TEXT........:

 _ EXCLUDE       WITHIN SYSTEM...: DCSYSTEM
                 VERSION NUMBER..: 99        _ HIGHEST    _ LOWEST
                     TEXT........:

 _ EXCLUDE       WITHIN SYSTEM...: DCSYSTEM
                 VERSION NUMBER..: 44        _ HIGHEST    _ LOWEST
                     TEXT........:
```

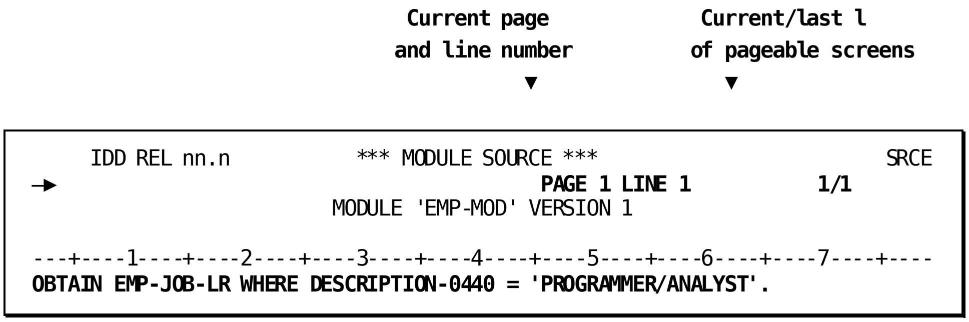
Free-form screens, such as the Comments (COMM) or Module Source (SRCE) screens, are similar to the free-form screens that the online compilers use.

The DDDL compiler considers each *line* of text to be one line of data.

Note: For more information, see Online Compilers.

Example

The following screen shows a sample of IDD Menu Facility free-form screen. The DDDL compiler considers the line beginning with OBTAIN to be one line of data.



Current page
and line number

Current/last l
of pageable screens

```
    IDD REL nn.n           *** MODULE SOURCE ***                     SRCE
→                                    PAGE 1 LINE 1          1/1
                               MODULE 'EMP-MOD' VERSION 1

---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----
OBTAIN EMP-JOB-LR WHERE DESCRIPTION-0440 = 'PROGRAMMER/ANALYST'.
```

Chapter 6: Using Top-line Commands

You can use top-line commands in an OCF, online compiler, or IDD Menu Facility session. These commands manipulate the contents of a work file, execute the statements in an OCF work file or execute the compiler, and suspend or terminate a session.

Note: For more information about editing commands that you enter in the input/output area of the screen, see Using Line Commands.

Entering Top-line Commands

To enter a top-line command, type the command on the top line of the screen and press Enter or use the program function (PF) key assigned to the desired command function (see Using Control Keys).

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

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

This section contains the following topics:

Summary of Top-line Commands

The following table gives short descriptions of the top-line commands.

Command	Description
APPLY	Updates the screen and work file but does not execute the statements in the OCF work file or execute the compiler.
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 top-line command and the associated PF key currently assigned to execute that command.
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 execution.
SWITCH (only if the development tool 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.

Command	Description
UPDATE	Updates the work file and executes the statements in the OCF work file or submits the work file to the compiler.

The rest of this chapter discusses each top-line command, its purpose, and its syntax.

APPLY

The APPLY command updates the screen and work file without executing the statements in the OCF work file or executing the compiler. This command is useful for reviewing edits before executing the statements in the OCF work file or submitting the work file to the compiler. 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 OCF work file or executing the compiler.
4. Make necessary corrections and issue the UPDATE command to execute the statements in the OCF work file or submit the work file to the compiler.

Syntax

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

CLEAR

The CLEAR command deletes all lines contained in the work file. Lines cleared from the work file cannot be restored. CLEAR is available with the OCF and all online compilers except the IDD Menu Facility.

Syntax

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

DELETE ALL

The DELETE ALL command (available only in the IDD Menu Facility) deletes all occurrences of information contained in all the pages of a pageable screen, and a free-form screen.

Syntax

```
►►─┬─ DELete ──┬─ ALL ──────────────────────────────────────────►◄
   ├─ EXClude ─┤
   └─ REMove ──┘
```

DISPLAY LINE

The DISPLAY LINE command displays a specified line from 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.

In the IDD Menu Facility, DISPLAY LINE works slightly differently for *pageable screens*. For these screens, the DDDL compiler considers each group of lines that represents a definition to be *one line* of data. DISPLAY LINE displays the first line of the definition at the top of the screen I/O area.

Note: DISPLAY LINE can be used in all pageable screens except the Record Elements (RELM) screen. This is because one RELM definition (viewed by the compiler as one line) is equal to one page. The DISPLAY PAGE command is used to display multiple occurrences of the Record Elements screen.

Syntax

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

Parameters

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. To avoid conflict with the PRINT command, PRIOR can only be abbreviated to PRIO. In the IDD Menu Facility, PRIOR can be abbreviated to PRI.

line-number

Displays the specified work file line.

+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.

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. You can enter any DISPLAY PAGE options without having to precede the option with the words DISPLAY PAGE. For example, you need only enter 12 to go to page 12 or LAST to go to the last page of the work file.

Syntax

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

Parameters

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.

To avoid conflict with the PRINT command, PRIOR can only be abbreviated to PRIO. In the IDD Menu Facility, PRIOR can be abbreviated to PRI.

page-number

Displays the specified page in the work file.

+page-count

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

-page-count

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

END

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

Syntax

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

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 ───┘
```

Parameters

UPDate

Updates the work file and executes the commands in the OCF work file or submits the work file to the compiler. UPDATE is the default.

APPly

Updates the work file but *does not* execute the commands in the OCF work file or execute the compiler.

ESCAPE

The ESCAPE command establishes the escape character you must enter as the first character of a line command. Line commands are discussed in more detail in Using Line Commands.

When specifying a line command, you type the current escape character in column 1 of the work file. The escape character signals that subsequent characters should be evaluated as a line command, rather than as an input statement or compiler input.

Syntax

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

Parameter

escape-character

Specifies any single alphanumeric character. The percent sign (%) is the default.

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.

In the IDD Menu Facility, FIND works slightly differently for *pageable screens*. For these screens, the line that contains the object string is not highlighted.

Syntax

```
►►── Find 'character-string' ─┬──────────┬──────────────────────────►◄
                              ├─ FWD ◄───┤
                              └─ BACk ───┘
```

Parameters

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.

BACk

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

HELP

The HELP command lists each top-line command and the PF key currently assigned to execute that command. To return to the work file, press any PF key or press ENTER.

In the IDD Menu Facility, HELP can be issued from any screen. This displays the Help tutorial related to the screen from which you issued the HELP command. Press PF1 to invoke the HELP command. To leave the Help tutorial and return to the prior screen, press Clear.

Syntax

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

INSERT

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

Syntax

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

You use INSERT differently depending on whether you are using the OCF and online compilers or the IDD Menu Facility as follows.

Using INSERT with the OCF and Online Compilers

To insert text into the work file during an OCF or online compiler session, do the following:

1. Issue the INSERT command, using one of the following methods:
   - *Top-line 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.

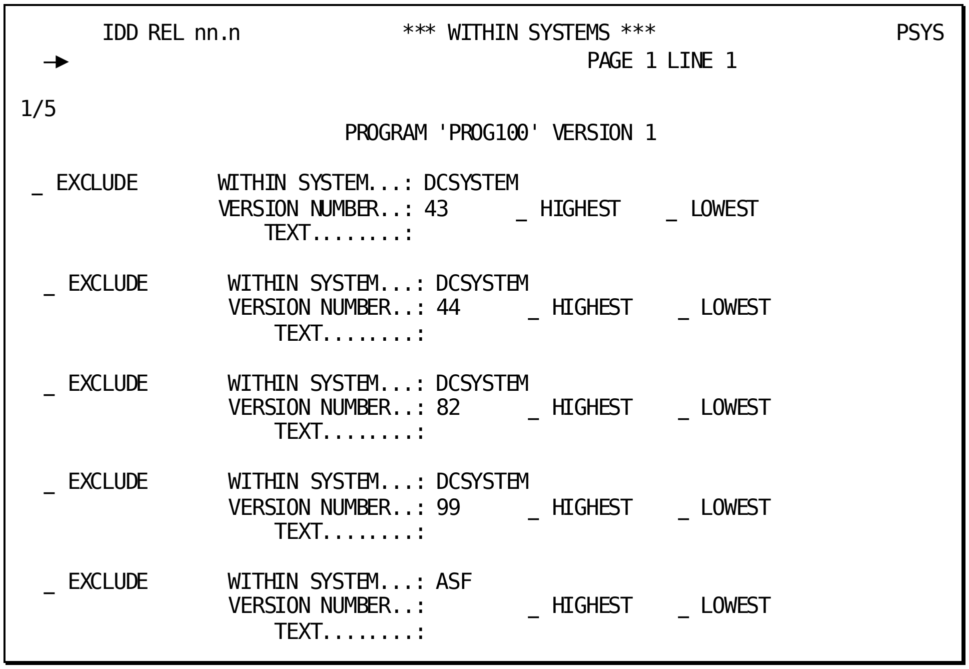
     The text editor inserts the new lines into the work file and updates the screen without executing the statements in the OCF work file or executing the compiler. To execute the statements or to submit the work file to the compiler, use the UPDATE command.
   - *UPDATE*—To execute the commands or execute the compiler, 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 OCF work file or compiles the contents of the updated work file.

Using INSERT with the IDD Menu Facility

Example

In the IDD Menu Facility, special considerations apply to the use of INSERT on *pageable screens*. The DDDL compiler considers each group of lines that represents a definition to be *one* line of data, as illustrated on the sample Program Within System (PSYS) screen in the IDD Menu Facility:



```
        IDD REL nn.n               *** WITHIN SYSTEMS ***                   PSYS
   →                                               PAGE 1 LINE 1

1/5
                                 PROGRAM 'PROG100' VERSION 1

 _ EXCLUDE        WITHIN SYSTEM...: DCSYSTEM
                  VERSION NUMBER..: 43       _ HIGHEST     _ LOWEST
                      TEXT........:

 _ EXCLUDE        WITHIN SYSTEM...: DCSYSTEM
                  VERSION NUMBER..: 44       _ HIGHEST     _ LOWEST
                      TEXT........:

 _ EXCLUDE        WITHIN SYSTEM...: DCSYSTEM
                  VERSION NUMBER..: 82       _ HIGHEST     _ LOWEST
                      TEXT........:

 _ EXCLUDE        WITHIN SYSTEM...: DCSYSTEM
                  VERSION NUMBER..: 99       _ HIGHEST     _ LOWEST
                      TEXT........:

 _ EXCLUDE        WITHIN SYSTEM...: ASF
                  VERSION NUMBER..:          _ HIGHEST     _ LOWEST
                      TEXT........:
```

- To insert text after all existing definitions:

  1. Display the last page of the pageable screen by scrolling forward or using the DISPLAY PAGE command.
  2. If there is a blank formatted area at the bottom of the screen, enter a new definition; otherwise, scroll forward to obtain a blank formatted area and enter a new definition.
  3. Continue issuing INSERT commands as each page is filled until you have entered all definitions.
  4. Press Enter or use the APPLY/UPDATE commands.

- To insert text before or between existing definitions:

  1. Determine the line that the new definition is to *precede*.
  2. Use a DISPLAY LINE command or scroll forward until the appropriate line is at the top of the screen.
  3. Type INSERT in the command area and press Enter.
  4. Enter text on any or all of the blank lines.
  5. Continue issuing INSERT commands as each page is filled until you have entered all definitions.
  6. Press Enter or use the APPLY/UPDATE commands.

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. The PRINT command is valid for CA IDMS/DC and CA IDMS UCF users only and is not available in the IDD Menu Facility.

Syntax

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

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. In the IDD Menu Facility, you can use the REPEAT command only on *free-form screens* such as the Comments (COMM) or Module Source (SRCE) screens.

Syntax

```
►►─── REPeat ─┬────────────────┬─────────────────────────────────────►◄
              └─ repeat-count ─┘
```

Parameters

REPeat

Repeats the specified line once.

repeat-count

Represents the number of times to repeat the line.

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.

In the IDD Menu Facility, the maximum value for *repeat-count* is 18.

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.

RESHOW is available with the OCF and all online compilers except the IDD Menu Facility. However, in the IDD Menu Facility, PA2 performs the same function.

Syntax

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

SUSPEND

The SUSPEND command allows you to exit from a session without ending the session. The current session is suspended, and control is returned 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.

Note: For more information about the Transfer Control Facility, see Transfer Control Facility.

The SUSPEND command can be issued from any screen in the IDD Menu Facility.

For CA IDMS ASF Users

If you are using ASF in the same online session, do not use SUSPEND to exit the schema or subschema compilers. Use the END or the SIGNOFF command.

Syntax

```
▶▶─── SUSpend ──────────────────────────────────────────────────────▶◀
```

SWAP

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

Swapping screens does not revoke the results of the previous execution or compilation. Any definitions that were successfully added to the data dictionary before you used the SWAP command remain in the dictionary after you use the command.

The SWAP command is available with the OCF and all online compilers except the IDD Menu Facility.

Syntax

```
▶▶─── SWAp ─────────────────────────────────────────────────────────▶◀
```

SWITCH

The SWITCH command allows you to transfer directly from one development tool to another under the Transfer Control Facility (TCF). For example, while under TCF you can use SWITCH to transfer from IDD to ADSC without having to return to the TCF Selection Screen. Use SWITCH to perform any of the following operations:

- Initiate a new development tool session.
- Resume a suspended development tool session.
- Display the TCF Selection Screen.
- Terminate or suspend a TCF session and return control to the DC/UCF system.

Note: For more information about the TCF, see Transfer Control Facility.

Syntax

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

Parameters

SWItch

Suspends the current development tool session. Control transfers back to the most recently suspended development tool session. If no suspended session exists, SWITCH transfers control to the TCF Selection Screen.

product-code

Specifies the development tool to which control should transfer.

Product-code identifies a development tool by a unique system-supplied invocation name defined at system generation on the TASK statement.

For example, IDD is the product code for online IDD; SSC is the product code for the online subschema compiler.

Product codes are defined for use only within TCF.

task-code

Identifies a development tool by its unique installation-defined invocation name.

For example, IDDT has been defined as the task code for online IDD; SSCT has been defined as the task code for the online subschema compiler. Task codes are defined at system generation time and can vary from site to site.

OLD

OLD, the default, resumes the most recently suspended session of the specified development tool. If no suspended session exists, OLD starts a new session of the development tool.

NEW

Starts a new session of the specified development tool.

Note: If you have a suspended IDD menu facility, online IDD, schema compiler, subschema compiler, or system generation compiler session, NEW resumes that suspended session. This is because you can have only one suspended session of each of these tools.

Note: The OLD and NEW options are not valid with the ADSA, ADSC, and MAPC task codes.

tcf-task-code

Specifies the task code that invokes the TCF. When you specify it with the SWITCH command, the TCF Selection Screen displays.

Tcf-task-code must be defined at system generation time and can vary from site to site. The default task code is TCF.

A sample TCF Selection Screen is shown at the end of these SWITCH command syntax rules.

CLEar and CLR

Suspends the development tool session, ends the TCF session (clearing the list of suspended sessions and DB defaults), and returns control to DC/UCF.

Your suspended development tool sessions are not affected by this action. You can resume any of your suspended sessions by invoking the appropriate development tool (while under TCF or directly from the DC/UCF system) and specifying the name of the definition being created in the suspended session. These suspended sessions exist until you sign off from the host TP monitor or the DC/UCF system terminates.

Note: The CLEar and CLR options are not used with ADSAT, ADSCT, or MAPCTtask codes.

SUSpend

Suspends the development tool session, suspends the TCF session (saving the list of suspended sessions and DB defaults), and returns control to DC/UCF.

Note: The ADSA, ADSC, and MAPC compilers automatically suspend sessions (and save the contents of the session in queue records) when you leave the compiler. Therefore, the SUSPEND option has no affect when used with ADSA, ADSC, or MAPC.

Your list will be available to you next time you invoke TCF. The list of suspended sessions is saved by TCF when you sign off from the host TP monitor or the DC/UCF system terminates.

You can resume any of your suspended development tool sessions by selecting the session from your list of suspended sessions (while under TCF). You also can resume any suspended session by invoking the appropriate development tool (while under TCF or directly from the DC/UCF system) and specifying the name of the definition being created in the suspended session.

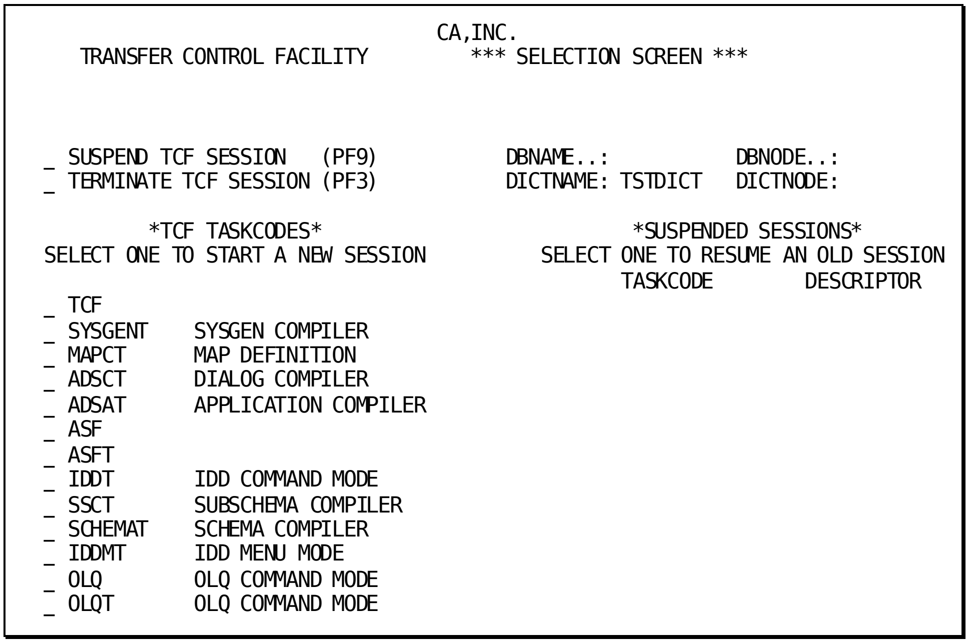
Note: Do not use SUSPEND to exit the schema or subschema compilers if you are using the Automatic System Facility (ASF) in the same online session. In this case, use END or the signoff command.

Example

You display the TCF Selection Screen by specifying the SWITCH *tcf-task-code*.

The TCF Selection Screen provides the following information:

- A list of suspended sessions. From this list you can select a session to be restarted.
- A list of all development tools available under TCF. From this list you can start a new session.
- A list of *database and data dictionary defaults* (DB defaults) that includes dbname, dictname, nodename, and dictnode. From this list you can set session defaults for these four options.



```
                                     CA,INC.
   TRANSFER CONTROL FACILITY            *** SELECTION SCREEN ***



 _ SUSPEND TCF SESSION   (PF9)         DBNAME..:          DBNODE..:
 _ TERMINATE TCF SESSION (PF3)         DICTNAME: TSTDICT  DICTNODE:

          *TCF TASKCODES*                        *SUSPENDED SESSIONS*
 SELECT ONE TO START A NEW SESSION          SELECT ONE TO RESUME AN OLD SESSION
                                                   TASKCODE        DESCRIPTOR
 _ TCF
 _ SYSGENT    SYSGEN COMPILER
 _ MAPCT      MAP DEFINITION
 _ ADSCT      DIALOG COMPILER
 _ ADSAT      APPLICATION COMPILER
 _ ASF
 _ ASFT
 _ IDDT       IDD COMMAND MODE
 _ SSCT       SUBSCHEMA COMPILER
 _ SCHEMAT    SCHEMA COMPILER
 _ IDDMT      IDD MENU MODE
 _ OLQ        OLQ COMMAND MODE
 _ OLQT       OLQ COMMAND MODE
```

Product and Task Code Usage

In a SWITCH command, you can specify a development tool's product or task code to transfer to that development tool. Additionally, you can use a task code defined for use under TCF to invoke the associated development tool directly from a DC/UCF system.

The following table lists product codes and sample task codes for use under TCF.

Development Tool	Product Code	Sample Task Code[1]	Site Task Code
CA ADS application compiler (ADSA)	ADSA	ADSAT	

Development Tool	Product Code	Sample Task Code[1]	Site Task Code
CA ADS dialog compiler (ADSC)	ADSC	ADSCT	
ASF	ASF	ASFT	
IDD menu facility	IDDM	IDDMT	
Online IDD	IDD	IDDT	
Online map compiler (MAPC)	MAPC	MAPCT	
Online command facility (OCF)	OCF	OCFT	
CA OLQ	OLQ	OLQT	
Online schema compiler	SCHEMA	SCHEMAT	
Online subschema compiler	SSC	SSCT	
Online system generation compiler	SYSGEN	SYSGENT	

1. Task codes for use under TCF are defined at system generation time and can vary from site to site.

SWITCH Command Specification Outcomes

Use SWITCH to transfer from one development tool session to another new or suspended development tool session, to the TCF Selection Screen, or back to the DC/UCF system.

The following table shows the outcome for each SWITCH specification.

SWITCH command specification	If you have suspended session(s)	If you do not have suspended session(s)
SWITCH	Resumes most recently suspended development tool session	Displays TCF Selection Screen
SWITCH *product or task code* OLD	Resumes most recently suspended session of requested tool	Starts new session of requested tool if tool is IDD or online compiler or displays DC/UCF screen if tool is ADSA, ADSC, or MAPC

SWITCH command specification	If you have suspended session(s)	If you do not have suspended session(s)
SWITCH *product or task code* NEW	Resumes most recently suspended session of requested tool (IDD and online compilers only)	Starts new session of requested tool, if possible
SWITCH *tcf-task-code*	Displays TCF Selection Screen	Displays TCF Selection Screen
SWITCH CLEAR	Clears list of suspended development tool sessions and displays DC/UCF screen	Displays DC/UCF screen
SWITCH SUSPEND	Saves list of suspended development tool sessions and displays DC/UCF screen	

Using the SWITCH Command

How to Specify the SWITCH Command

Specify the SWITCH command on the command line in a given development tool. For example, specify SWITCH as a top-line command in IDD.

When using ADSA, ADSC, or MAPC, you can also select the SWITCH activity from the action bar of the main menu.

Using SWITCH with Other Development Tools

When operating under TCF, you can specify SWITCH and any optional parameters in any development tool presented in this table.

The following table describes methods for specifying SWITCH in development tools under TCF.

Development Tool	Specifying the SWITCH Command	Specifying Optional Parameters
ADSA, ADSC, MAPC	Select the SWITCH activity from the action bar of the main menu or Type SWITCH in the command line.	■ Specify a task or product code on the TASK ID field of the SWITCH activity. ■ Specify a task code or product code as a SWITCH parameter on the command line. Example: switch idd

Development Tool	Specifying the SWITCH Command	Specifying Optional Parameters
IDD	Type SWITCH in the command line.	■ Specify a product code or task code as a SWITCH parameter. ■ Specify OLD or NEW as a parameter after a product or task code. Example: switch sysgen old
OLQ	Type SWITCH in the command line.	■ Specify a task code as a SWITCH parameter Example: switch schema
Online schema, subschema, and system generation compilers, and the OCF	Type SWITCH in the command line.	■ Specify a product code, task code, or keyword as a SWITCH parameter. ■ Specify OLD or NEW as a parameter after a product or task code. Example: switch idd new

UPDATE

The UPDATE command applies your most recent changes to the work file *and* executes the commands in the OCF work file or submits the contents of the work file to the compiler. In contrast, you use the APPLY command if you only want to update the work file.

Syntax

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

Chapter 7: Using Line Commands

Line commands are editor commands that you can use to edit a work file on OCF screens or online compiler editor screens. For example, you repeat a line by typing the *repeat* line command on the actual line to be repeated.

This section contains the following topics:

Overview

You use line commands to copy, delete, move, and repeat lines or blocks of lines within a work file. Additionally, the top line command allows you to reposition the screen by moving a selected line to the top of the screen.

This chapter provides the following information about line commands:

- A summary of available line commands
- Considerations for entering line commands
- How to use block commands

Used Only on Free-form Screens

In the IDD Menu Facility, you can use line commands only on *free-form screens*, such as the Comments (COMM) or Module Source (SRCE) screens.

Escape Character

Line commands must begin with the *escape character* in column 1. This manual uses the default escape character, the percent sign (%). You can define a different escape character by using the ESCAPE command.

Note: For more information about the ESCAPE command, see Chapter 6:.

Available Commands

The following table lists line commands, their function, and the format in which you enter them. The string **(space)** shown throughout this chapter represents the number of spaces you must enter following each line command:

Function	Single command format	Block command format
After (Line identifier)	////////////// //////////////	**%A**(space)
Before (Line identifier)	////////////// //////////////	**%B**(space)
Copy Lines	**%C**[**n**](space)	**%CB**(space). . .**%CE**(space)
Delete Lines	**%D**[**n**](space)	**%DB**(space). . .**%DE**(space)
Move Lines	**%M**[**n**](space)	**%MB**(space). . .**%ME**(space)
Repeat Lines	**%R**[**n**](space)	**%RE**(space)
Top Line	**%T**(space)	**%T**(space)

What to Consider

- Line commands cannot contain embedded spaces, but you must enter a space after them, as follows:

```
  Line one
  Line two
%r(space)Line three
```

- The text editor processes line commands when you press Enter or any PF key. When line commands are present in the file, pressing a control key *does not* invoke OCF or the compiler (even if ENTER=UPDATE is in effect).

What Can You Do?

You can perform the following functions, each of which is discussed in this section, using line commands:

- Copy lines or blocks of lines
- Delete lines or blocks of lines
- Enter line commands for blocks of lines
- Move lines or blocks of lines
- Repeat lines or blocks of lines
- Reposition lines on the screen

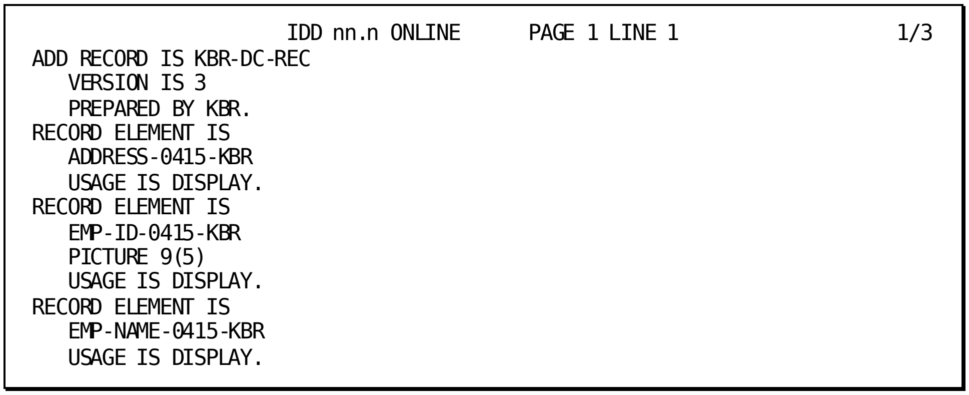
Copying Lines

To copy a line or a block of lines to another place in the work file, use any of the following forms of the copy (C) line command:

Use:	To:
%C(space)	Copy the indicated line to the destination marked by the %A or %B command.
%Cn(space)	Copy the specified number of lines to the destination marked by the %A or %B command. For example, %C5 copies the line on which the command is located and the four lines immediately following that line.
%CB(space). . .**%CE**(space)	Copy a block of lines to the destination marked by the %A or %B command. The two lines marked with %CB and %CE and all intervening lines are copied. The original lines are not deleted. **Note**: You can enter block commands (%CB and %CE), so that they span more than one page of the work file.
%A(space) or **%B**(space)	Determine where the selected lines are copied: ■ **%A**(space) identifies the line *after* which the targeted lines are copied. ■ **%B**(space) identifies the line *before* which the targeted lines are copied.

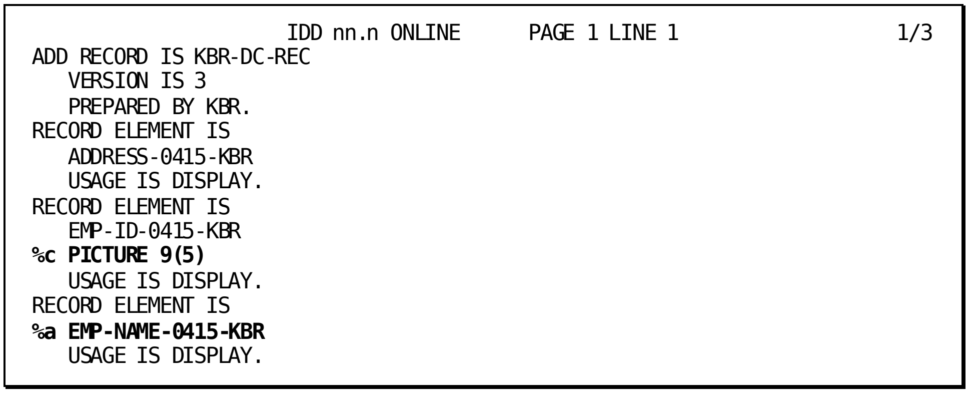
Scenario—Copying Lines

The following scenario illustrates how to copy lines on the online compiler screens. The following example shows the original work file as it appears on the online compiler screen.



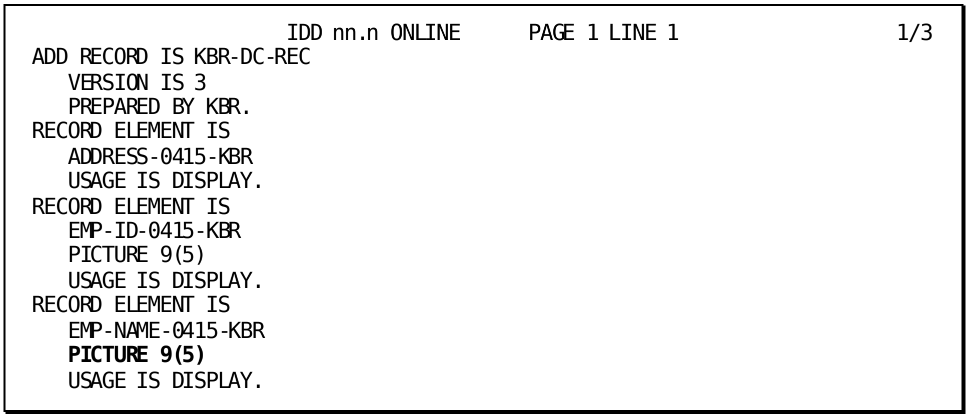
```
                        IDD nn.n ONLINE      PAGE 1 LINE 1                  1/3
ADD RECORD IS KBR-DC-REC
   VERSION IS 3
   PREPARED BY KBR.
RECORD ELEMENT IS
   ADDRESS-0415-KBR
   USAGE IS DISPLAY.
RECORD ELEMENT IS
   EMP-ID-0415-KBR
   PICTURE 9(5)
   USAGE IS DISPLAY.
RECORD ELEMENT IS
   EMP-NAME-0415-KBR
   USAGE IS DISPLAY.
```

The copy (%c) line command copies the COBOL picture for EMP-ID-0415 KBR to the line after the line designated by the copy after (%a) command.



```
                        IDD nn.n ONLINE      PAGE 1 LINE 1                  1/3
ADD RECORD IS KBR-DC-REC
   VERSION IS 3
   PREPARED BY KBR.
RECORD ELEMENT IS
   ADDRESS-0415-KBR
   USAGE IS DISPLAY.
RECORD ELEMENT IS
   EMP-ID-0415-KBR
%c PICTURE 9(5)
   USAGE IS DISPLAY.
RECORD ELEMENT IS
%a EMP-NAME-0415-KBR
   USAGE IS DISPLAY.
```

The following online compiler screen displays the results of the copy procedure in the modified work file:



```
                      IDD nn.n ONLINE      PAGE 1 LINE 1                   1/3
ADD RECORD IS KBR-DC-REC
   VERSION IS 3
   PREPARED BY KBR.
RECORD ELEMENT IS
   ADDRESS-0415-KBR
   USAGE IS DISPLAY.
RECORD ELEMENT IS
   EMP-ID-0415-KBR
   PICTURE 9(5)
   USAGE IS DISPLAY.
RECORD ELEMENT IS
   EMP-NAME-0415-KBR
   PICTURE 9(5)
   USAGE IS DISPLAY.
```

Deleting Lines

To delete a line or block of lines, use any of the following forms of the delete (D) command:

Use:	To:
%D(space)	Delete the indicated line.
%Dn(space)	Delete the specified number of lines. For example, %D5 deletes the command line and the four lines immediately following it.
%DB(space) . . . **%DE**(space)	Delete a block of lines. The two lines marked with %DB and %DE and all intervening lines are deleted. **Note**: You can enter %DB/%DE so that it spans more than one page of the work file.

Entering Block Commands

What to Consider

You can use line commands to modify a block of definition lines at one time. The following considerations apply when you use line commands for a block of lines:

- You use block commands to perform an operation on two or more *consecutive* lines. For example, you can move three consecutive lines by using the block **move** command, as follows:

  Sample input:

  ```
  %mb(space)This is the beginning of the text block
      This is the middle of the text block
  %me(space)This is the end of the text block
   %a(space)This is the line after which you move text
  ```

  Sample results:

  ```
  This is the line after which you move text
  This is the beginning of the text block
  This is the middle of the text block
  This is the end of the text block
  ```

- You can enter line commands in any order. For example, you can specify the target line (%A or %B) either before or after indicating the line or block of lines. After you have entered all the line commands for the specified operation, you perform the operation by pressing Enter.
- The editor displays the literal BLK (block) in the message area of the screen after you have entered a block command. The editor continues to display this literal as a reminder until you either complete or cancel the current copy, delete, move, or replace operation.

Common Text Editing Errors

The text editor assumes that you made an error when you enter conflicting commands. In this case, the editor nullifies all the line commands that you entered. The following table demonstrates text editing errors:

Type of error	Example
Mixing block commands	**%CB**(space) (copy begin) . . . **%ME**(space) (move end)

Type of error	Example
Specifying a target line inside a block	**%CB**(space) . . **%A**(space)◀ This is a target inside a block to be copied **%CE**(space)
Specifying the start of the block after the end of the block	**%CE**(space)◀ This is the end of a block . . . **%CB**(space)◀ This is the beginning of the block

Moving Lines

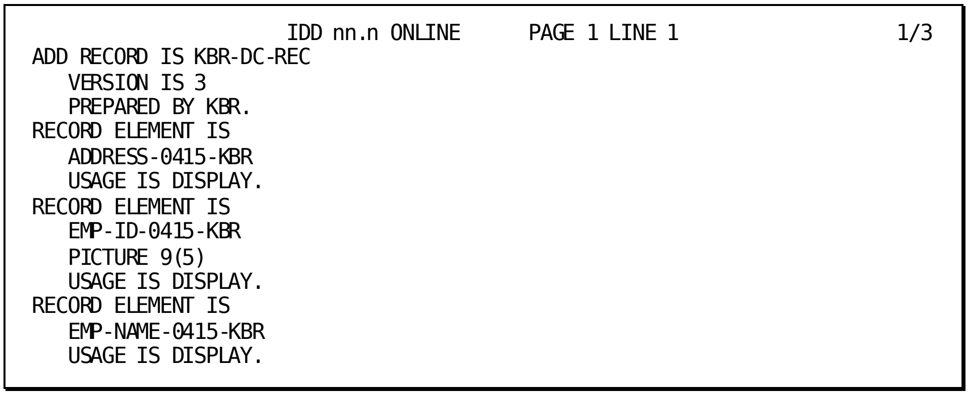
To move a line or a block of lines to another place in the work file, use any of the following line commands:

Use:	To:
%M(space)	Move the indicated line to the destination marked by the %A (after) or %B (before) command
%Mn(space)	Move the specified number of lines to the destination marked by the %A (after) or %B (before) command. For example, %M5 moves the line on which the command is located and the four lines immediately following it.
%MB(space). . .**%ME**(space)	Move a block of lines to the destination marked by the %A (after) or %B (before) command. The two lines marked with %MB and %ME and all intervening lines are moved. **Note**: You can enter %MB...%ME so that it spans more than one page of the work file. For an example of moving lines, see Entering Block Commands.
%A(space) or **%B**(space)	■ **%A**(space)identifies the line *after* which the targeted lines are moved. ■ **%B**(space) identifies the line *before* which the targeted lines are moved.

Scenario—Moving Lines

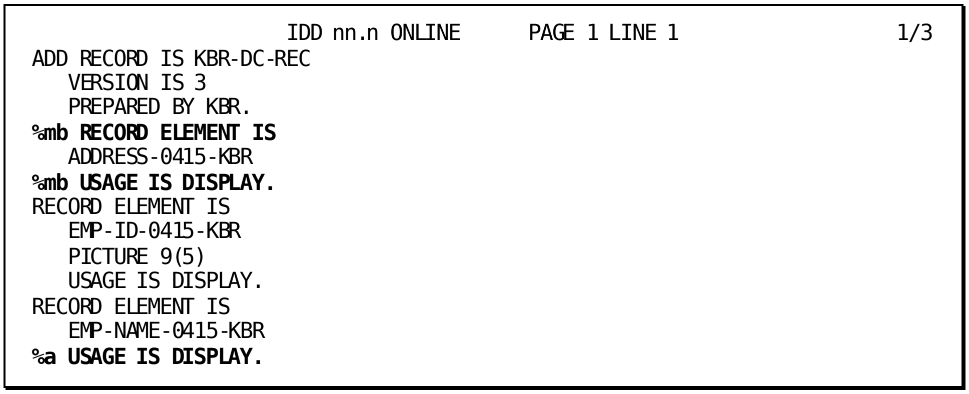
The following scenario illustrates moving a block of text using line commands. Note that you type the line commands over existing characters, and that you type a space at the end of each command.

The following sample screen shows the original work file as it appears on the online compiler screen.



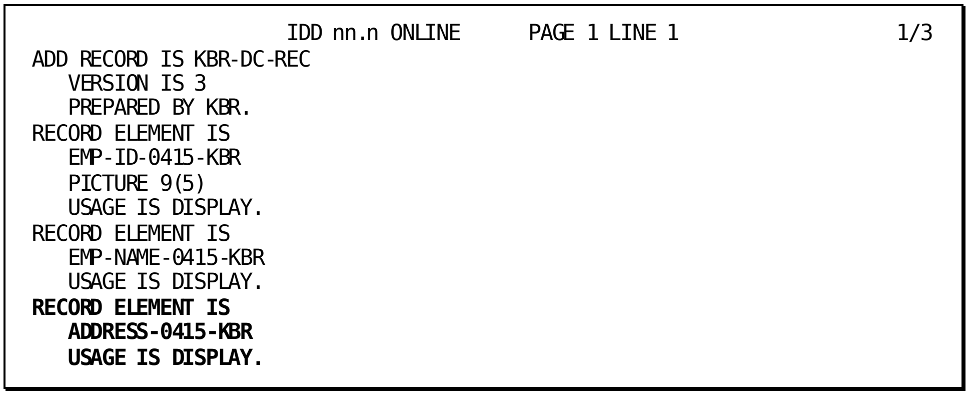
```
                     IDD nn.n ONLINE     PAGE 1 LINE 1                 1/3
ADD RECORD IS KBR-DC-REC
   VERSION IS 3
   PREPARED BY KBR.
RECORD ELEMENT IS
   ADDRESS-0415-KBR
   USAGE IS DISPLAY.
RECORD ELEMENT IS
   EMP-ID-0415-KBR
   PICTURE 9(5)
   USAGE IS DISPLAY.
RECORD ELEMENT IS
   EMP-NAME-0415-KBR
   USAGE IS DISPLAY.
```

The move begin (%mb) and move end (%me) block commands specify the text block you want to move. The move after (%a) command identifies the line after which you want to place the text block.



```
                        IDD nn.n ONLINE      PAGE 1 LINE 1                   1/3
ADD RECORD IS KBR-DC-REC
   VERSION IS 3
   PREPARED BY KBR.
%mb RECORD ELEMENT IS
   ADDRESS-0415-KBR
%mb USAGE IS DISPLAY.
RECORD ELEMENT IS
   EMP-ID-0415-KBR
   PICTURE 9(5)
   USAGE IS DISPLAY.
RECORD ELEMENT IS
   EMP-NAME-0415-KBR
%a USAGE IS DISPLAY.
```

The following sample screen shows the results of the block move procedure.



```
                        IDD nn.n ONLINE      PAGE 1 LINE 1                   1/3
ADD RECORD IS KBR-DC-REC
   VERSION IS 3
   PREPARED BY KBR.
RECORD ELEMENT IS
   EMP-ID-0415-KBR
   PICTURE 9(5)
   USAGE IS DISPLAY.
RECORD ELEMENT IS
   EMP-NAME-0415-KBR
   USAGE IS DISPLAY.
RECORD ELEMENT IS
   ADDRESS-0415-KBR
   USAGE IS DISPLAY.
```

Repeating Lines

To repeat a line or block of lines, use any of the following forms of the repeat (R) line command:

Use:	To:
%R(space)	Repeats the indicated line one time.
%Rn(space)	Repeat the indicated line *n* times. For example, %R2 repeats the line two times.
%RBn(space) . . . **%RE**(space)	Repeat a block of lines *n* times. %RB*n* defines the beginning of the block and the number of times the block is to be repeated. %RE defines the end of the block. **Note**: The %R line command is not subject to the same repetition limit as the top-line command REPEAT. For example, %R100 *is* a valid command, REPEAT 100 *is not*.

Scenario—Repeating Lines

The following sample screens illustrate the process of repeating lines. The following screen shows the original work file:

```
                      IDD nn.n ONLINE      PAGE 1 LINE 1                 1/3
02 WK-GROUP USAGE IS DISPLAY OCCURS 5 TIMES.
   05 WK-VARIABLE-A USAGE IS DISPLAY PIC X(2).
   05 WK-BIT-1 USAGE IS BIT PIC X(3).
```

To repeat a line, enter %r2 on the line you want to repeat.

```
                      IDD nn.n ONLINE      PAGE 1 LINE 1                 1/3
02 WK-GROUP USAGE IS DISPLAY OCCURS 5 TIMES.
   05 WK-VARIABLE-A USAGE IS DISPLAY PIC X(2).
%r2 5 WK-BIT-1 USAGE IS BIT PIC X(3).
```

After entering the repeat command, the repeated lines appear so that you can modify them in the work file.

```
                      IDD nn.n ONLINE      PAGE 1 LINE 1                 1/5
02 WK-GROUP USAGE IS DISPLAY OCCURS 5 TIMES.
   05 WK-VARIABLE-A USAGE IS DISPLAY PIC X(2).
   05 WK-BIT-1 USAGE IS BIT PIC X(3).
   05 WK-BIT-1 USAGE IS BIT PIC X(3).
   05 WK-BIT-1 USAGE IS BIT PIC X(3).
```

Repositioning Lines on the Screen

Use the top (T) line command to reposition the lines currently displayed on the screen. This command has only one format:

%T(space)

When you enter this command at a specific line, the text editor moves the work file forward until the line specified by the %T is the top line of the screen.

Chapter 8: Using Control Keys

You can use control keys (such as PF keys, PA keys, and Enter) as an alternative to top-line commands. Default control key assignments differ between the OCF, online compilers, and IDD Menu Facility. To display the current control key assignments, use the HELP command. Top-line and line commands override PF keys. When you enter a top-line or line command, PF keys have the same function as Enter, which updates the work file. PA1, PA2, and Clear have priority over the commands in the command area. If you press one of these keys, no data is transmitted from the terminal.

This section contains the following topics:

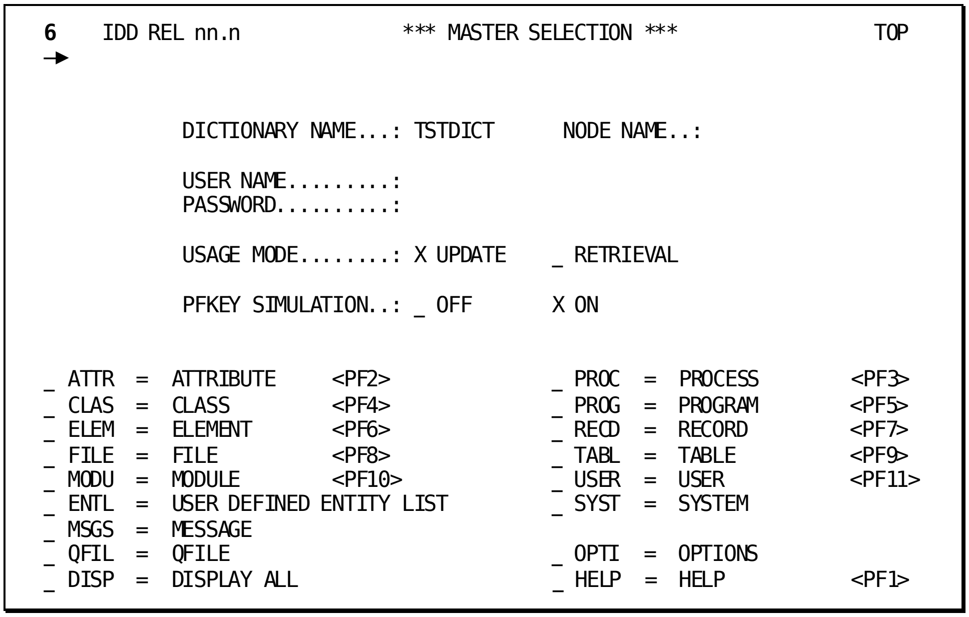
Simulating PF Keys

On terminals that have no PF keys, you can perform PF key-related functions in the IDD Menu Facility by using *PF key simulation*. To activate PF-key simulation in the IDD Menu Facility:

1. Enter any character in the PF KEY SIMULATION ON response field on the Master Selection screen.
2. Enter a one- or two-digit PF key number in the simulated PF key field in the upper left corner of the screen. Note that the HOME key moves the cursor to the simulated PF key field when PF key simulation is in effect.
3. Press Enter.

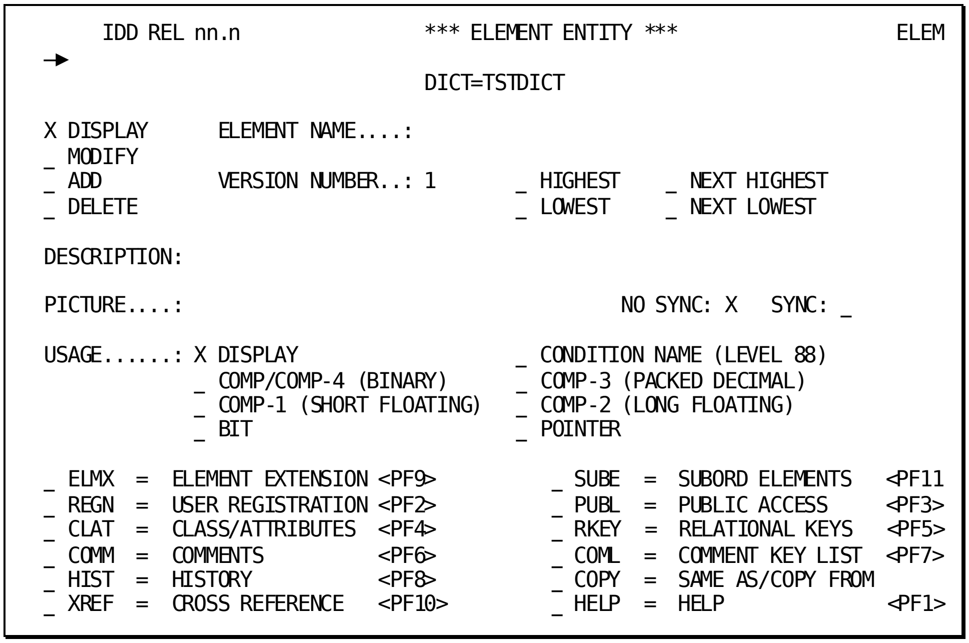
Example

On the following Master Selection screen, you enter 6 in the PF KEY SIMULATION ON field to simulate PF6, which requests the Element Entity screen:



```
6    IDD REL nn.n              *** MASTER SELECTION ***               TOP
→

            DICTIONARY NAME...: TSTDICT      NODE NAME..:

            USER NAME.........:
            PASSWORD..........:

            USAGE MODE........: X UPDATE     _ RETRIEVAL

            PFKEY SIMULATION..: _ OFF        X ON


_ ATTR  =  ATTRIBUTE     <PF2>               _ PROC  =  PROCESS       <PF3>
_ CLAS  =  CLASS         <PF4>               _ PROG  =  PROGRAM       <PF5>
_ ELEM  =  ELEMENT       <PF6>               _ RECD  =  RECORD        <PF7>
_ FILE  =  FILE          <PF8>               _ TABL  =  TABLE         <PF9>
_ MODU  =  MODULE        <PF10>              _ USER  =  USER          <PF11>
_ ENTL  =  USER DEFINED ENTITY LIST          _ SYST  =  SYSTEM
_ MSGS  =  MESSAGE
_ QFIL  =  QFILE                             _ OPTI  =  OPTIONS
_ DISP  =  DISPLAY ALL                       _ HELP  =  HELP          <PF1>
```

After entering 6 on the Master Selection screen, you access the Element Entity screen.



```
        IDD REL nn.n                   *** ELEMENT ENTITY ***                ELEM
 →
                                      DICT=TSTDICT

 X DISPLAY      ELEMENT NAME....:
 _ MODIFY
 _ ADD          VERSION NUMBER..: 1         _ HIGHEST    _ NEXT HIGHEST
 _ DELETE                                   _ LOWEST     _ NEXT LOWEST

 DESCRIPTION:

 PICTURE....:                                        NO SYNC: X   SYNC: _

 USAGE......: X DISPLAY                     _ CONDITION NAME (LEVEL 88)
              _ COMP/COMP-4 (BINARY)        _ COMP-3 (PACKED DECIMAL)
              _ COMP-1 (SHORT FLOATING)     _ COMP-2 (LONG FLOATING)
              _ BIT                         _ POINTER

 _ ELMX  =  ELEMENT EXTENSION <PF9>         _ SUBE  =  SUBORD ELEMENTS   <PF11
 _ REGN  =  USER REGISTRATION <PF2>         _ PUBL  =  PUBLIC ACCESS     <PF3>
 _ CLAT  =  CLASS/ATTRIBUTES  <PF4>         _ RKEY  =  RELATIONAL KEYS   <PF5>
 _ COMM  =  COMMENTS          <PF6>         _ COML  =  COMMENT KEY LIST  <PF7>
 _ HIST  =  HISTORY           <PF8>         _ COPY  =  SAME AS/COPY FROM
 _ XREF  =  CROSS REFERENCE   <PF10>        _ HELP  =  HELP              <PF1>
```

Using OCF and Online Compiler Control Keys

The following table lists the installation default control keys, their associated top-line commands, and their functions in the OCF and online compilers. The table also provides space for you to associate site-specific control keys with the OCF and online compiler functions.

Press (default)	Site key	To invoke	Which
PF1, PF13 PF8, PF20		DISPLAY PAGE NEXT	Scrolls forward one page.
PF2, PF14 PF7, PF19		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		APPLY	Updates screen contents and work file but does not invoke the compiler.
PF6, PF18		UPDATE	Updates work file and executes the compiler.
PF9, PF21		SWAP	Restores the work file contents.

Press (default)	Site key	To invoke	Which
PF12, PF24		PRINT	Prints the work file contents.
PA1		Cancel FIND	Cancels the FIND command.
PA2		RESHOW	Cancels changes to the current screen and redisplays the screen.
Clear		CLEAR	Clears the contents of the work file.
Enter		=APPLY =UPDATE	Updates the screen and the work file. Updates the work file and executes the compiler.

Using IDD Menu Facility Global Control Keys

The following table lists global control keys for the IDD Menu Facility. The table does not include local control keys, which are available only on specific IDD Menu Facility screens.

Press	To invoke	Which
PF1, PF13	HELP	Displays online help tutorial associated with the current screen.
PF7, PF19	DISPLAY PAGE PRIOR	Scrolls backward one page (pageable and free-form screens only).
PF8, PF20	DISPLAY PAGE NEXT	Scrolls forward one page (pageable and free-form screens only).
PA1	Restart	Erases information typed on the screen since the last control key was pressed.
PA2	Refresh screen	Cancels changes to the current screen and redisplays the screen.
Clear	Prior screen	Displays the prior screen.
Enter	=APPLY =UPDATE	Updates the screen and the work file. Updates the work file and executes the compiler.

Chapter 9: Transfer Control Facility

The Transfer Control Facility (TCF) allows you to transfer from one CA IDMS online development tool to another without having to return first to CA IDMS/DC (that is, without having to sign off from the first tool and sign on to the next).

This section contains the following topics:

Transfer Control Facility Overview

The TCF simplifies transfer between development tools by passing signon and session information between tools.

Use TCF to Transfer Directly to Another Tool

Using TCF, you can perform the following tasks:

1. Suspend a session of a development tool
2. Transfer directly to another development tool
3. Then transfer back to the suspended session of the first development tool and resume the session.

For example, if you have forgotten the exact names of dictionary records to be copied into a schema, you can temporarily suspend your online schema compiler session while you use the online Data Dictionary (IDD). After verifying the record names, you can resume work in the schema compiler where you left off.

Tools You Can Invoke from TCF

Development tools and compilers invoked while you are using TCF are said to operate *under* TCF. The following development tools and compilers can be used under TCF and are discussed in this document:

- Automatic System Facility (ASF)
- CA ADS
- CA OLQ
- IDD menu facility
- Online command facility
- Online IDD
- Online map compiler
- Online schema compiler
- Online subschema compiler
- Online system generation compiler

Each development tool is established separately for use under TCF at system generation time. TCF configuration can vary from site to site. For more information about how development tools are established under TCF, see the *CA IDMS System Generation Guide*.

TCF Keeps Track of Your Sessions

TCF automatically keeps track of your suspended development tool sessions. You can view your current list of suspended sessions at any time by displaying the TCF Selection Screen.

SWITCH Command

You can transfer directly from one development tool to another by using the SWITCH command. SWITCH also allows you to transfer to the Selection Screen. With SWITCH and the Selection Screen you can initiate a new development tool session or resume a suspended one.

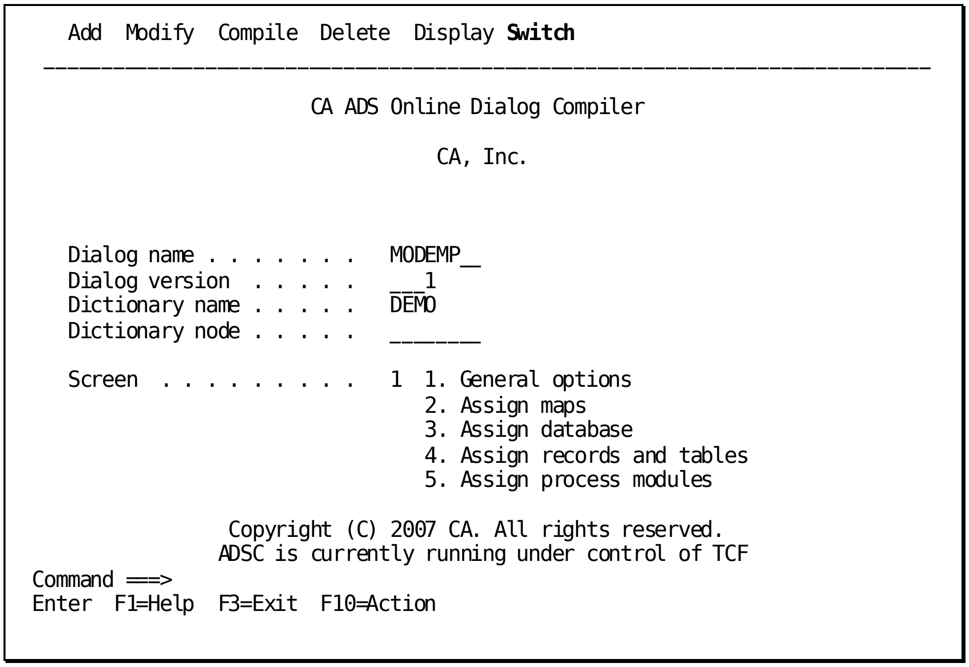
Note: For more information about the SWITCH command, see SWITCH.

Examples

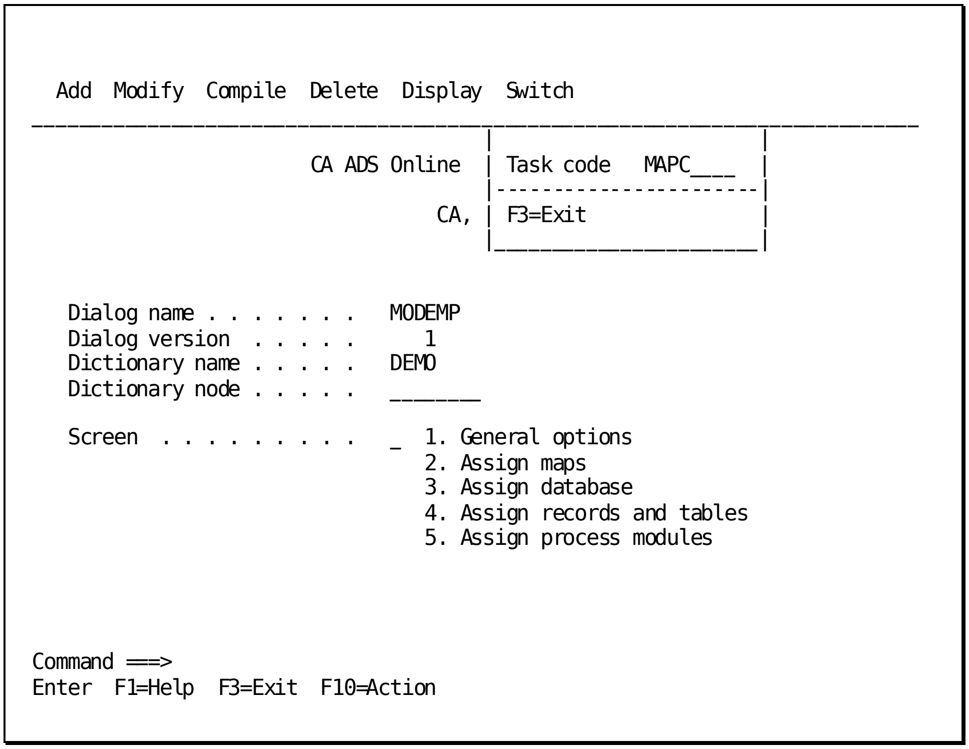
You can transfer to and from development tools under TCF at any time.

Transferring from ADSC to the Online Map Compiler

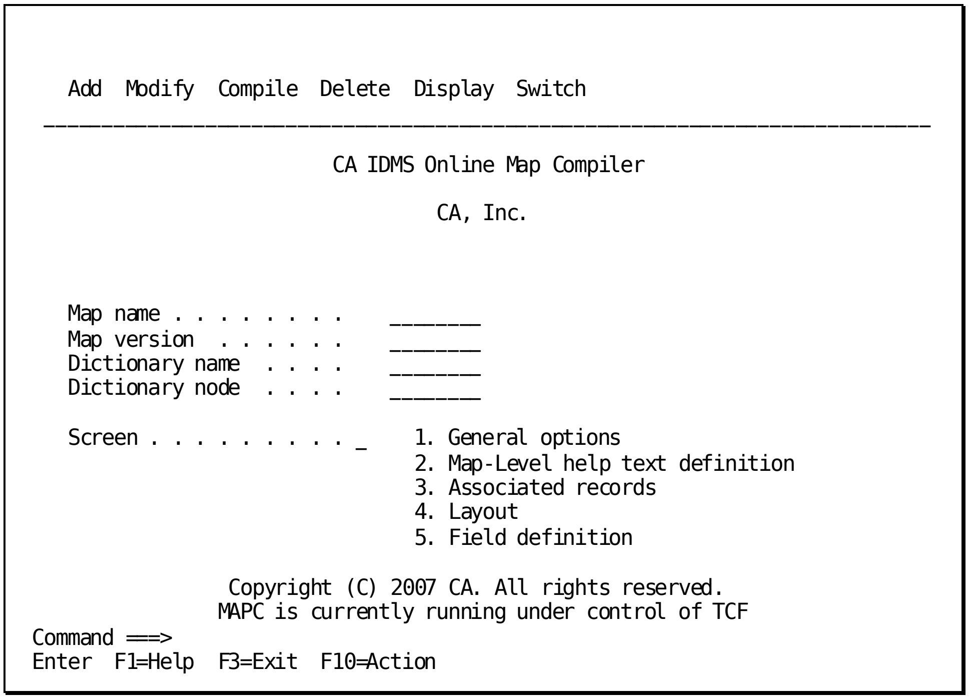
For example, while using ADSC to define a dialog, you remember that the related map definition is still incomplete. You can suspend your ADSC session and transfer directly to MAPC as shown in the following screen:



```
   Add  Modify  Compile  Delete  Display Switch
 ------------------------------------------------------------------------------

                      CA ADS Online Dialog Compiler

                                CA, Inc.



   Dialog name . . . . . . .   MODEMP__
   Dialog version  . . . . .   ___1
   Dictionary name . . . . .   DEMO
   Dictionary node . . . . .   ________

   Screen  . . . . . . . . .   1  1. General options
                                  2. Assign maps
                                  3. Assign database
                                  4. Assign records and tables
                                  5. Assign process modules

                 Copyright (C) 2007 CA. All rights reserved.
                ADSC is currently running under control of TCF
Command ===>
Enter  F1=Help  F3=Exit  F10=Action
```



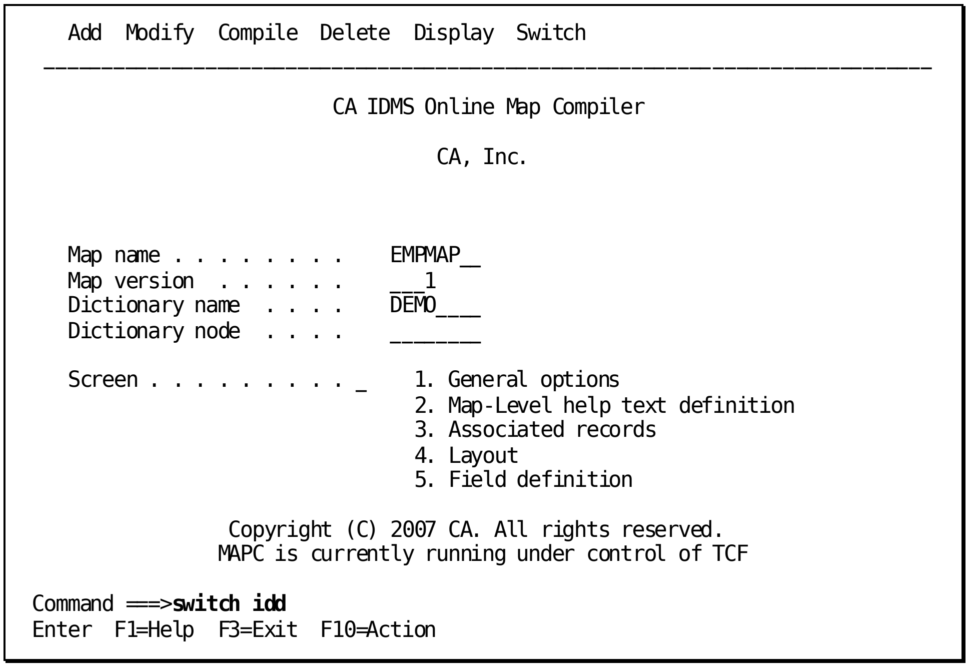
```
  Add  Modify  Compile  Delete  Display  Switch
____________________________________________________________________________
                                       |                      |
                         CA ADS Online | Task code  MAPC____  |
                                       |----------------------|
                                   CA, | F3=Exit              |
                                       |______________________|


   Dialog name . . . . . . .   MODEMP
   Dialog version  . . . . .      1
   Dictionary name . . . . .   DEMO
   Dictionary node . . . . .   ________

   Screen  . . . . . . . . .   _  1. General options
                                  2. Assign maps
                                  3. Assign database
                                  4. Assign records and tables
                                  5. Assign process modules




Command ===>
Enter  F1=Help  F3=Exit  F10=Action
```



```
   Add  Modify  Compile  Delete  Display  Switch
 ____________________________________________________________________________

                          CA IDMS Online Map Compiler

                                   CA, Inc.



   Map name . . . . . . . .    ________
   Map version  . . . . . .    ________
   Dictionary name  . . . .    ________
   Dictionary node  . . . .    ________

   Screen . . . . . . . . . _    1. General options
                                 2. Map-Level help text definition
                                 3. Associated records
                                 4. Layout
                                 5. Field definition

                 Copyright (C) 2007 CA. All rights reserved.
                MAPC is currently running under control of TCF
Command ===>
Enter  F1=Help  F3=Exit  F10=Action
```

Transferring from MAPC to IDD

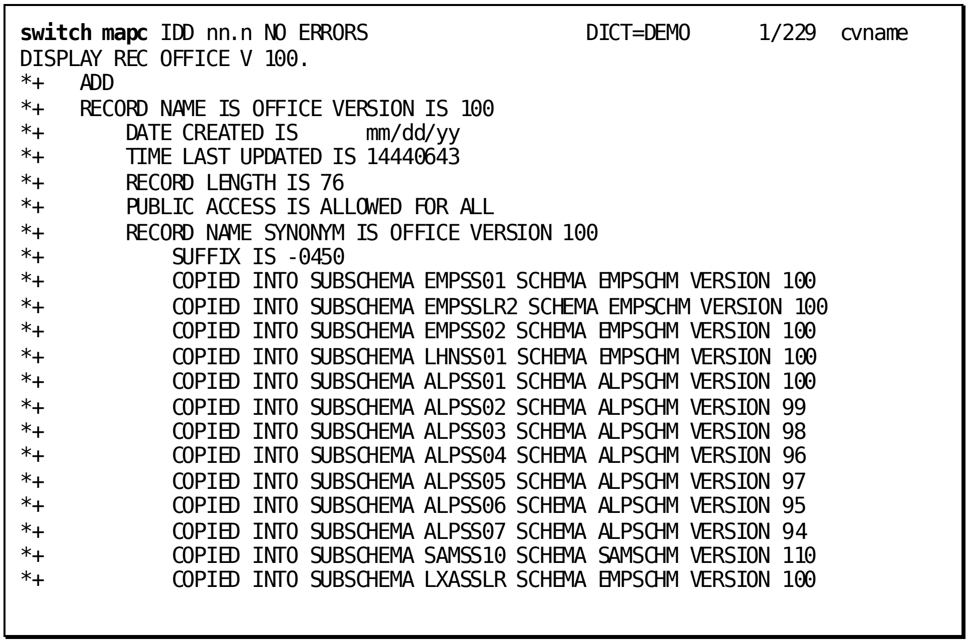
While using MAPC to complete your map, you need to verify information about a record associated with the map. You can use the SWITCH command to suspend MAPC and transfer directly to online IDD to view the record definition.



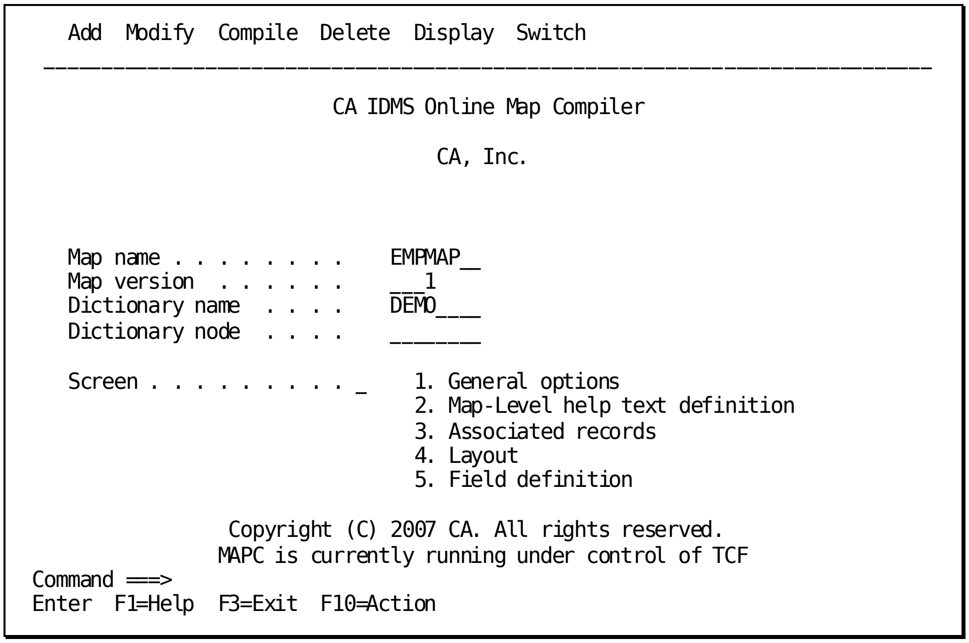
```
   Add  Modify  Compile  Delete  Display  Switch
 ---------------------------------------------------------------------------

                         CA IDMS Online Map Compiler

                                  CA, Inc.



   Map name . . . . . . . .    EMPMAP__
   Map version  . . . . . .    ___1
   Dictionary name  . . . .    DEMO____
   Dictionary node  . . . .    ________

   Screen . . . . . . . . . _    1. General options
                                 2. Map-Level help text definition
                                 3. Associated records
                                 4. Layout
                                 5. Field definition

                Copyright (C) 2007 CA. All rights reserved.
               MAPC is currently running under control of TCF

 Command ===>switch idd
 Enter  F1=Help  F3=Exit  F10=Action
```

Transferring from IDD back to MAPC

After using IDD to view the record definition, you can use the SWITCH command to transfer directly back to MAPC.



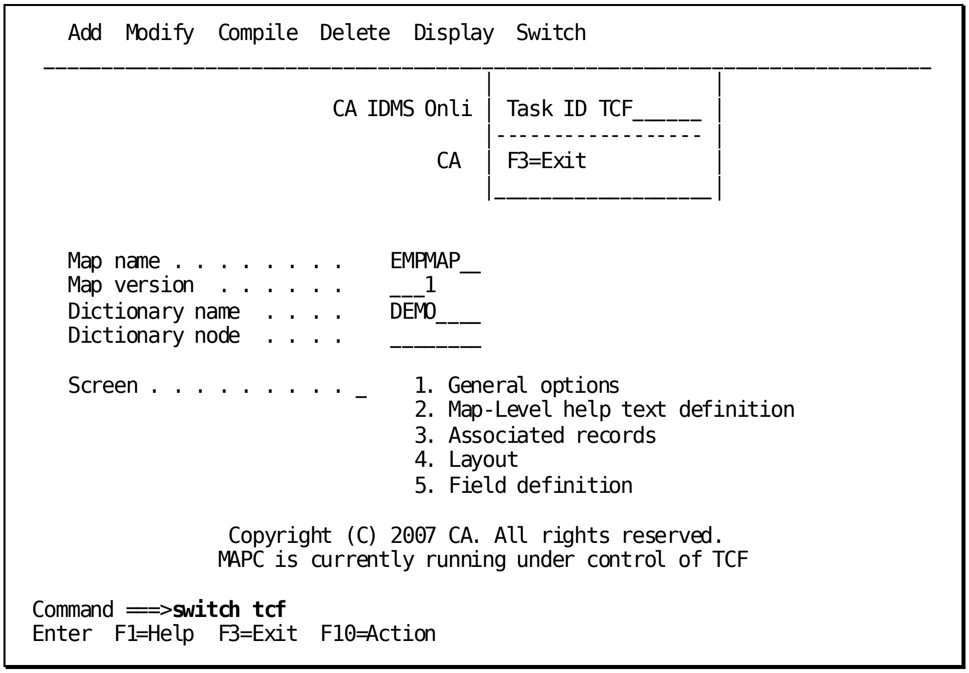
```
switch mapc IDD nn.n NO ERRORS                  DICT=DEMO      1/229   cvname
DISPLAY REC OFFICE V 100.
*+   ADD
*+   RECORD NAME IS OFFICE VERSION IS 100
*+       DATE CREATED IS      mm/dd/yy
*+       TIME LAST UPDATED IS 14440643
*+       RECORD LENGTH IS 76
*+       PUBLIC ACCESS IS ALLOWED FOR ALL
*+       RECORD NAME SYNONYM IS OFFICE VERSION 100
*+           SUFFIX IS -0450
*+           COPIED INTO SUBSCHEMA EMPSS01 SCHEMA EMPSCHM VERSION 100
*+           COPIED INTO SUBSCHEMA EMPSSLR2 SCHEMA EMPSCHM VERSION 100
*+           COPIED INTO SUBSCHEMA EMPSS02 SCHEMA EMPSCHM VERSION 100
*+           COPIED INTO SUBSCHEMA LHNSS01 SCHEMA EMPSCHM VERSION 100
*+           COPIED INTO SUBSCHEMA ALPSS01 SCHEMA ALPSCHM VERSION 100
*+           COPIED INTO SUBSCHEMA ALPSS02 SCHEMA ALPSCHM VERSION 99
*+           COPIED INTO SUBSCHEMA ALPSS03 SCHEMA ALPSCHM VERSION 98
*+           COPIED INTO SUBSCHEMA ALPSS04 SCHEMA ALPSCHM VERSION 96
*+           COPIED INTO SUBSCHEMA ALPSS05 SCHEMA ALPSCHM VERSION 97
*+           COPIED INTO SUBSCHEMA ALPSS06 SCHEMA ALPSCHM VERSION 95
*+           COPIED INTO SUBSCHEMA ALPSS07 SCHEMA ALPSCHM VERSION 94
*+           COPIED INTO SUBSCHEMA SAMSS10 SCHEMA SAMSCHM VERSION 110
*+           COPIED INTO SUBSCHEMA LXASSLR SCHEMA EMPSCHM VERSION 100
```



```
   Add  Modify  Compile  Delete  Display  Switch
 ----------------------------------------------------------------------------

                        CA IDMS Online Map Compiler

                                CA, Inc.



   Map name . . . . . . . .    EMPMAP__
   Map version  . . . . . .    ___1
   Dictionary name  . . . .    DEMO____
   Dictionary node  . . . .    ________

   Screen . . . . . . . . . _    1. General options
                                 2. Map-Level help text definition
                                 3. Associated records
                                 4. Layout
                                 5. Field definition

                 Copyright (C) 2007 CA. All rights reserved.
                MAPC is currently running under control of TCF
Command ===>
Enter  F1=Help  F3=Exit  F10=Action
```

Transferring to TCF from MAPC

After you generate the map, you want to transfer to the TCF Selection Screen to see what options are currently available to you. You can transfer to the TCF Selection Screen from MAPC by entering 'switch tcf' on the command line or by selecting the SWITCH option from the action bar of the online map compiler main menu.

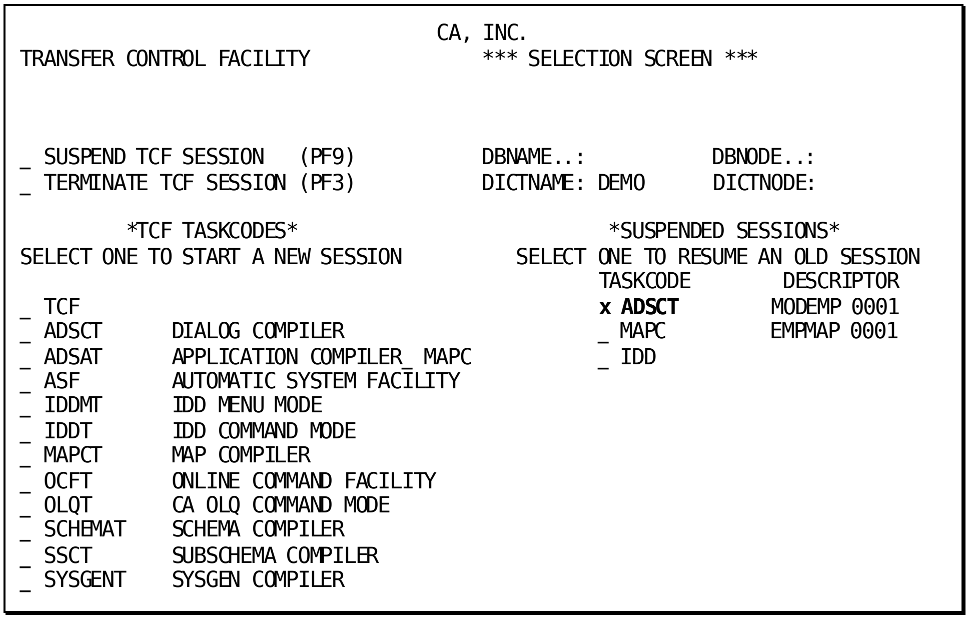


```
   Add  Modify  Compile  Delete  Display  Switch
 _____________________________________________________________________________
                                      |                   |
                        CA IDMS Onli  | Task ID TCF______ |
                                      |------------------ |
                                 CA   | F3=Exit           |
                                      |___________________|


   Map name . . . . . . . .    EMPMAP__
   Map version  . . . . . .    ___1
   Dictionary name  . . . .    DEMO____
   Dictionary node  . . . .    ________

   Screen . . . . . . . . . _    1. General options
                                 2. Map-Level help text definition
                                 3. Associated records
                                 4. Layout
                                 5. Field definition

                 Copyright (C) 2007 CA. All rights reserved.
                MAPC is currently running under control of TCF

 Command ===>switch tcf
 Enter  F1=Help  F3=Exit  F10=Action
```

The Selection Screen Is Displayed

The Selection Screen is displayed when transferring from the online map compiler to TCF.

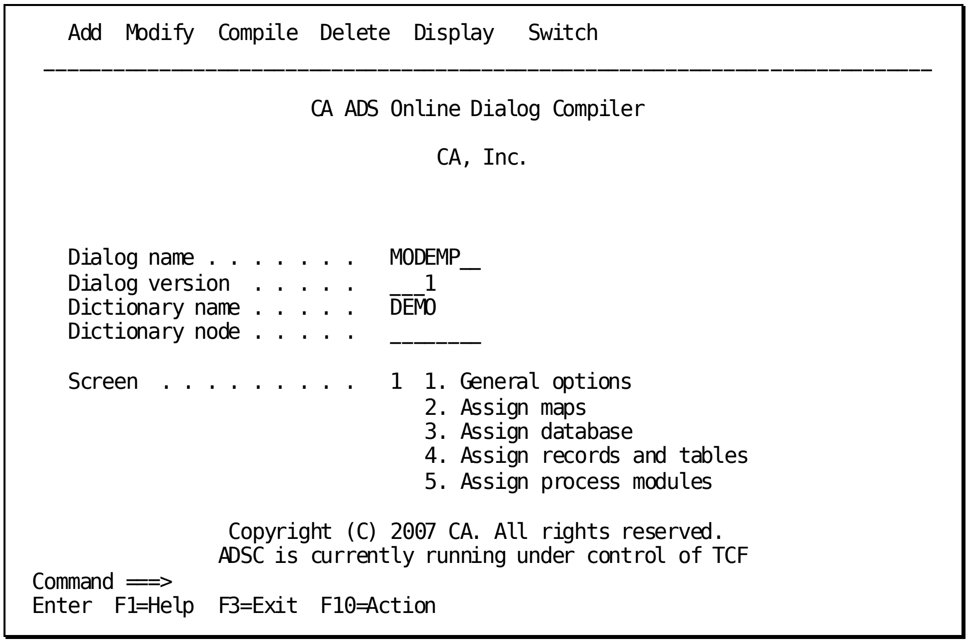
If you decide to go back and complete the MODEMP dialog definition, you can resume your suspended ADSC session by typing **x** in front of the session descriptor on the TCF Selection Screen.



```
                                 CA, INC.
TRANSFER CONTROL FACILITY            *** SELECTION SCREEN ***


_ SUSPEND TCF SESSION   (PF9)        DBNAME..:         DBNODE..:
_ TERMINATE TCF SESSION (PF3)        DICTNAME: DEMO    DICTNODE:

        *TCF TASKCODES*                       *SUSPENDED SESSIONS*
SELECT ONE TO START A NEW SESSION       SELECT ONE TO RESUME AN OLD SESSION
                                               TASKCODE       DESCRIPTOR
_ TCF                                          x ADSCT        MODEMP 0001
_ ADSCT      DIALOG COMPILER                   _ MAPC         EMPMAP 0001
_ ADSAT      APPLICATION COMPILER_ MAPC        _ IDD
_ ASF        AUTOMATIC SYSTEM FACILITY
_ IDDMT      IDD MENU MODE
_ IDDT       IDD COMMAND MODE
_ MAPCT      MAP COMPILER
_ OCFT       ONLINE COMMAND FACILITY
_ OLQT       CA OLQ COMMAND MODE
_ SCHEMAT    SCHEMA COMPILER
_ SSCT       SUBSCHEMA COMPILER
_ SYSGENT    SYSGEN COMPILER
```

Note: TCF uses system generation parameters to construct the Selection Screen dynamically, so the TCF Selection Screen displayed at your site may differ from the one shown. Furthermore, system administrators at each site can add other online applications to the TCF environment. For details about preparing a program to run under TCF, see the *CA IDMS Navigational DML Programming Guide*.

Your Suspended ADSC Session Is Resumed



```
    Add  Modify  Compile  Delete  Display   Switch
  ____________________________________________________________________________

                         CA ADS Online Dialog Compiler

                                   CA, Inc.



    Dialog name . . . . . . .   MODEMP__
    Dialog version  . . . . .   ___1
    Dictionary name . . . . .   DEMO
    Dictionary node . . . . .   ________

    Screen  . . . . . . . . .   1  1. General options
                                   2. Assign maps
                                   3. Assign database
                                   4. Assign records and tables
                                   5. Assign process modules

                  Copyright (C) 2007 CA. All rights reserved.
                 ADSC is currently running under control of TCF
 Command ===>
 Enter  F1=Help  F3=Exit  F10=Action
```

Chapter 10: Using TCF

To use TCF, you must first invoke TCF from a DC/UCF system. Control remains within TCF until you exit from TCF and return to DC/UCF.

This section contains the following topics:

Using the TCF

While you are under TCF, you can transfer directly between any of the development tools that operate under TCF without losing your place within the various tools. TCF also allows you to transfer to and from the Selection Screen at any time.

Invoking TCF from DC/UCF, transferring between development tools, invoking and exiting from a development tool, and exiting from TCF are discussed in this section.

Specifying a Task Code

Invoke TCF from a DC/UCF system by specifying any task code that is defined for use under TCF. For example, invoke TCF from CA IDMS/DC by specifying a task code (such as 'tcf') after the ENTER NEXT TASK CODE prompt, as follows:

```
ENTER NEXT TASK CODE:
 tcf
```

Note: A task code is the unique name (such as ADSAT) or the function key (such as PF3) that requests access to a development tool or online compiler. Task codes for use under TCF are defined at system generation time and can differ from site to site.

If you are using a teleprocessing monitor other than CA IDMS/DC, specify a task code after the prompt presented by that monitor.

Selecting a Task Code

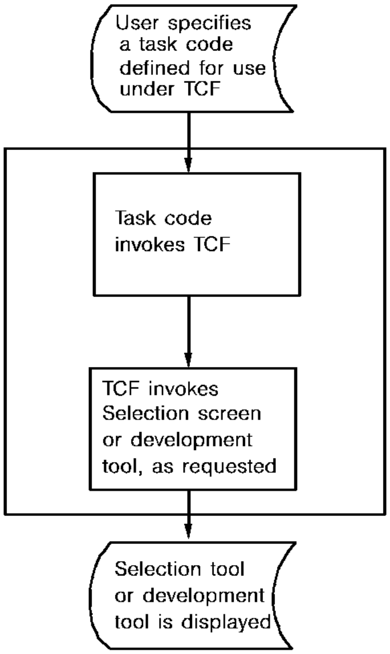
Decide which task code to use based on what you need to access under TCF. For example, you can use a task code to access either:

- **The TCF Selection Screen**—The Selection Screen displays and allows you to select options under TCF. A task code for the Selection Screen is defined at system generation time. For example, TCF can be defined as the task code for the Selection screen.
- **A development tool**—Each development tool under TCF is associated with a unique task code at system generation time. For example, IDDT can be defined as the task code for online IDD under TCF.

Invoking TCF from a DC/UCF System

TCF is invoked when you specify a task code defined for use under TCF. Then, TCF invokes the Selection Screen or the development tool identified by the supplied task code.

The following diagram shows what happens when you supply a task code to a DC/UCF system.


User specifies a task code defined for use under TCF
Task code invokes TCF
TCF invokes Selection screen or development tool, as requested
Selection tool or development tool is displayed

Transferring Between Development Tools

While using TCF, you can transfer control to and from development tools whenever necessary. TCF handles all your requests to invoke or exit from development tools and transfers control accordingly. How TCF evaluates your requests to invoke and exit from development tools is discussed next. For more information about using specific tools under TCF, see Development Tools under TCF.

Invoking a Development Tool

Invoking Tools from Different Locations

You can invoke a development tool under TCF from any of the following locations:

- *DC/UCF*—Specify the task code defined for the development tool under TCF.

  It is important to invoke tools using the task code defined for use under TCF at system generation time. For a list of sample task codes associated with specific product codes, see the product and task code table in SWITCH. For example, specify the task code defined for ADSC under TCF (such as ADSCT) to invoke ADSC directly from DC/UCF.

- *The TCF Selection Screen*—Select the development tool's task code from the screen.

  Select a task code from the Selection Screen by typing a nonblank character in front of the task code name (even if the task code is a function key name).

  For example, if SSCT is the task code for the online subschema compiler under TCF, invoke the subschema compiler by selecting SSCT from the list of task codes on the Selection Screen. For more information about the Selection Screen, see TCF Selection Screen.

- *Another development tool*—Use any command or function key that transfers control from that tool to another.

  For example, use the SWITCH command in online IDD to transfer from IDD to ADSC. For more information about transferring from specific tools under TCF, see Development Tools under TCF.

Beginning a New Session or Resuming a Session

When you invoke a development tool under TCF, you can request to do the following:

- *Begin a new session* of the tool.

  For example, start a new session of MAPC to define a new map. TCF handles a request to start a new development tool session as described in the table that follows.

- *Resume a suspended session* of the tool.

  For example, resume a suspended ADSC session to continue working on the dialog being defined in the requested ADSC session.

Requesting a New Development Tool Session

TCF handles your request to start a new development tool session based on the number of simultaneous sessions you can have for the requested tool.

Development tool	TCF response when you request a new development tool session
■ ADSA ■ ADSC ■ MAPC	Because you can have two or more simultaneous sessions of each of these tools, TCF starts a new session of the requested tool.
■ IDD menu facility ■ Online command facility ■ Online IDD ■ Online schema compiler ■ Online system generation compiler	Because you can have only one session at a time of each of these tools, TCF: ■ Resumes your suspended session of the requested tool, if you have one ■ Starts a new session of the requested tool if you do not currently have a suspended session of that tool
■ ASF ■ CA OLQ	Because you can have only one session at a time of each of these tools, TCF restarts a session using retained signon and table information.

Exiting from a Development Tool

If you specify a destination when you exit a tool, TCF transfers you to the requested location. You can request transfer to another development tool under TCF, the Selection Screen and DC/UCF.

If you do not specify a destination when you exit a tool, TCF transfers you to the most recently suspended development tool session. If you have no suspended sessions, TCF transfers you to the DC/UCF system.

You remain under TCF when you exit from a development tool session unless you explicitly request exit from TCF or you have no suspended TCF sessions. For more information about the transfer of control from a specific development tool, see the discussion of that development tool in Development Tools under TCF.

The current development tool session can be either suspended or terminated when you exit from the tool session.

Suspending a Session

If the development tool session is suspended, the current definition or report is saved.

For example, if you are using ADSC to define a dialog and need to use IDD to verify the name of a process module, you can suspend the current ADSC session to transfer to IDD. You can then resume the ADSC session when you are finished using IDD.

When you suspend a development tool session under TCF, an identifier for the session is added to the end of your list of suspended sessions. The identifier for the suspended development tool session remains in your list until you either resume the suspended session or request that your list be cleared on exit from TCF (as described next in Exiting from TCF).

You can view your *list of suspended sessions* at any time by transferring to the Selection Screen. For example, if you have forgotten the name of one of your suspended dialog definitions, you could transfer to the Selection screen and look at your list of suspended sessions. You also can resume any of the listed sessions by selecting the session from your list on the Selection Screen. For more information about the Selection Screen and your list of suspended sessions, see TCF Selection Screen chapter.

Terminating a Session

If the development tool session is terminated, the current definition or report is cleared.

For example, you can terminate your current IDD session on exit if you are finished using IDD to verify information.

Exiting from TCF

Exiting from TCF returns control to the DC/UCF system. You can exit from either of the following locations:

- *From the Selection Screen*, as discussed in TCF Selection Screen chapter.
- *From a development tool* under TCF, by using the SWITCH command as discussed in SWITCH.

Suspending or Terminating TCF When You Exit

You can either suspend or terminate TCF when you exit. Decide whether to suspend or terminate TCF as follows:

- *Suspend TCF* to save your current list of suspended development tool sessions. Your list will be available the next time you invoke TCF.

  You can resume any of your suspended sessions by selecting the session from your list of suspended sessions (while under TCF). You can also resume any suspended session by invoking the appropriate development tool (while under TCF or directly from DC/UCF) and specifying the name of the definition being created in the suspended session.

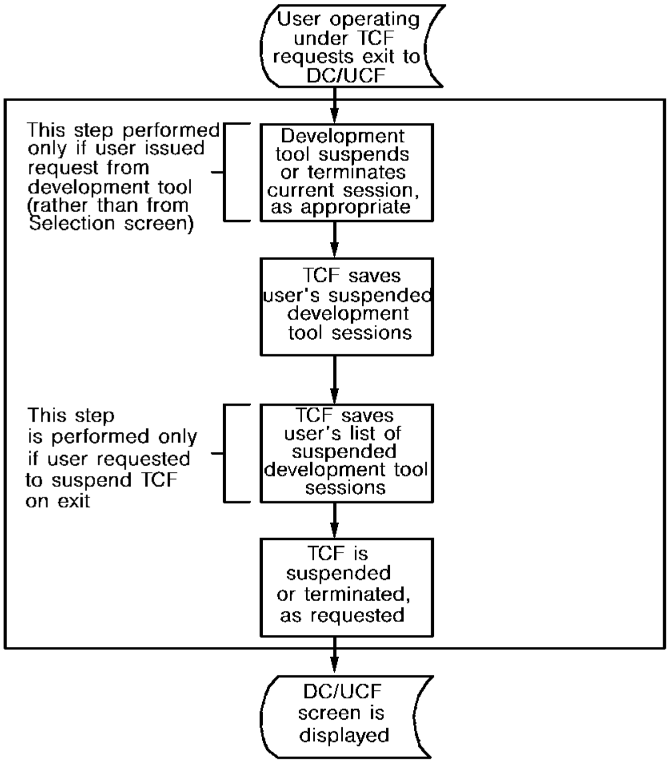
- *Terminate TCF* to clear your current list of suspended development tool sessions. Your suspended sessions are not affected by this action.

  You can resume any of your suspended sessions by invoking the appropriate development tool (while under TCF or directly from DC/UCF) and specifying the name of the definition being created.

Exiting from TCF to the DC/UCF System

You can exit to the DC/UCF system from the Selection Screen or directly from a development tool. If the user suspends TCF on exit, TCF saves the user's list of suspended development tool sessions. If the user terminates TCF on exit, TCF clears the user's list of suspended development tool sessions. In both cases, the actual suspended development tool sessions are saved.

The following diagram shows what happens when you exit from TCF.


User operating under TCF requests exit to DC/UCF
This step performed only if user issued request from development tool (rather than from Selection screen)
Development tool suspends or terminates current session, as appropriate
TCF saves user's suspended development tool sessions
This step is performed only if user requested to suspend TCF on exit
TCF saves user's list of suspended development tool sessions
TCF is suspended or terminated, as requested
DC/UCF screen is displayed

Chapter 11: TCF Selection Screen

The Selection Screen lists and allows you to select options that are currently available to you under TCF. Use the Selection Screen to perform any of the following operations.

- View your list of suspended development tool sessions
- Specify a dictionary or database name to be passed between development tools
- Initiate a new development tool session
- Resume a suspended development tool session
- Return control to the DC/UCF system

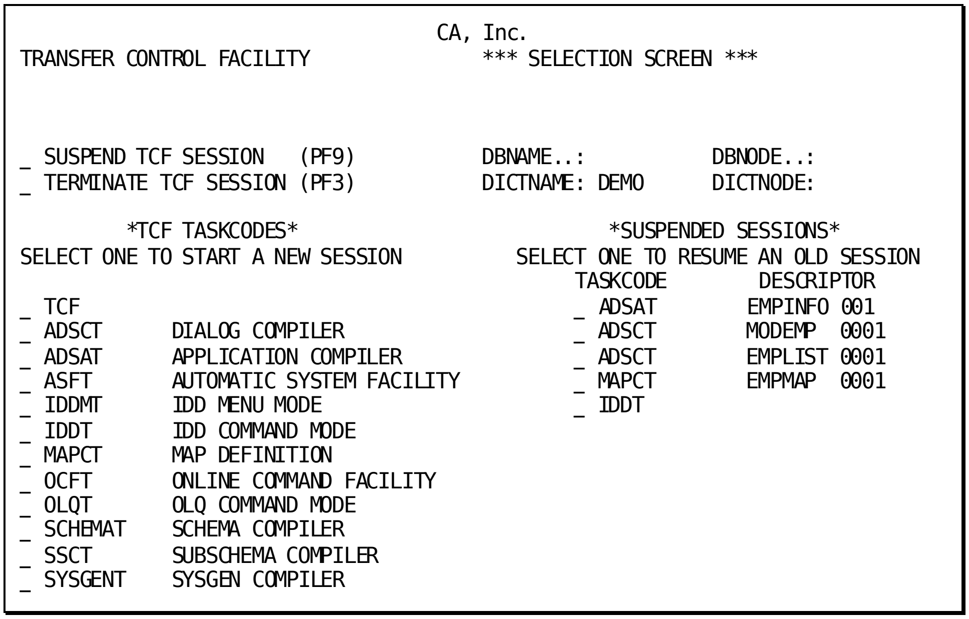
Transferring to the Selection Screen

You can transfer to the Selection Screen at any time while under TCF. The Selection Screen lists the task codes of all available development tools. Additionally, the Selection Screen displays an up-to-date list of your suspended development tool sessions each time you transfer to the Selection Screen. The Selection Screen can display identifiers for up to 13 suspended development tool sessions.

Note: When you have 13 suspended sessions, TCF automatically transfers control to the Selection Screen after you suspend each additional development tool session. A message on the Selection Screen informs you that your list is full, and that the suspended session's identifier has not been added to the list. To resume the additional suspended session, invoke a new session of the tool and specify the name of the definition being created in the suspended session.

Sample Selection Screen

The Selection Screen allows you to transfer control to a new development tool session, a suspended development tool session, or back to the DC/UCF system. The following screen depicts a sample Selection Screen: Options available from the Selection Screen are described in the table that follows. Sample identifiers made up of the session's task code and descriptor are shown.



```
                                    CA, Inc.
TRANSFER CONTROL FACILITY               *** SELECTION SCREEN ***


_ SUSPEND TCF SESSION   (PF9)           DBNAME..:           DBNODE..:
_ TERMINATE TCF SESSION (PF3)           DICTNAME: DEMO      DICTNODE:

         *TCF TASKCODES*                          *SUSPENDED SESSIONS*
SELECT ONE TO START A NEW SESSION          SELECT ONE TO RESUME AN OLD SESSION
                                                TASKCODE        DESCRIPTOR
_ TCF                                         _ ADSAT          EMPINFO 001
_ ADSCT      DIALOG COMPILER                  _ ADSCT          MODEMP  0001
_ ADSAT      APPLICATION COMPILER             _ ADSCT          EMPLIST 0001
_ ASFT       AUTOMATIC SYSTEM FACILITY        _ MAPCT          EMPMAP  0001
_ IDDMT      IDD MENU MODE                    _ IDDT
_ IDDT       IDD COMMAND MODE
_ MAPCT      MAP DEFINITION
_ OCFT       ONLINE COMMAND FACILITY
_ OLQT       OLQ COMMAND MODE
_ SCHEMAT    SCHEMA COMPILER
_ SSCT       SUBSCHEMA COMPILER
_ SYSGENT    SYSGEN COMPILER
```

TCF Selection Screen Options

Option	Field	Description
Exit from TCF	SUSPEND TCF SESSION (PF9)	■ Returns control to DC/UCF. ■ Saves your list of suspended development tool sessions.
	TERMINATE TCF SESSION (PF3)	■ Returns control to DC/UCF. ■ Clears your list of suspended development tool sessions.

Option	Field	Description
	USAGE: Select either exit option in one of the following ways: ■ Type a nonblank character in front of either field. For example, type an 'x' to select SUSPEND TCF SESSION as shown in the following code: x SUSPEND TCF SESSION (PF9) ■ Press the associated function key. For example, press PF9 to select SUSPEND TCF SESSION.	
Specify a database or dictionary name	DBNAME	Names the database (if any) that you have most recently specified.
	DBNODE	Names the database node (if any) that you have most recently specified.
	DICTNAME	Names the dictionary (if any) that you have most recently specified.
	DICTNODE	Names the dictionary node (if any) that you have most recently specified.
	USAGE: Displays and allows you to specify the database or dictionary name to be passed between development tools. For example, specify the DEMO dictionary as shown in the following code: DICTNAME: demo DICTNODE: You also can specify database or dictionary values in either of the following locations: ■ A development tool ■ DC/UCF (using a DCUF command)	
Start a new development tool session	TCF	Task code for TCF Selection Screen

Option	Field	Description
	ADSCT	Task code for ADSC
	ADSAT	Task code for ADSA
	ASFT	Task code for ASF
	IDDMT	Task code for IDD menu facility
	IDDT	Task code for online IDD
	MAPCT	Task code for MAPC
	OCFT	Task code for online command facility
	OLQT	Task code for OLQ
	SCHEMAT	Task code for online schema compiler
	SSCT	Task code for online subschema compiler
	SYSGENT	Task code for system generation compiler
	USAGE: Request a new development tool session by typing a nonblank character in front of its task code. For example, select online IDD from the Selection Screen by typing an 'x' as shown in the following code: x IDDT IDD COMMAND MODE	
Resume a suspended development tool session	TASKCODE	Lists the task code for each session in your list of suspended sessions. For example, the task code for ADSC (such as ADSCT) is listed for each of your suspended ADSC sessions.
	DESCRIPTOR	List descriptors for your suspended ADSA, ADSC, and MAPC sessions. A descriptor is created from the name and version number of the entity being defined in the suspended session.

Option	Field	Description
	USAGE: Displays your list of suspended development tool sessions by task code (and by descriptor when appropriate). Resume a suspended session by typing a nonblank character in front of the session identifier. For example, resume a suspended ADSC session by typing an 'x' as shown in the following code: x ADSCT EMPLIST 0001 In the preceding example, the suspended ADSC session defines the EMPLIST dialog.	

Note: A different task code may be defined for use under TCF at your site.

Chapter 12: Development Tools under TCF

Development tools are established for use under TCF at system generation time. Any of the following development tools and compilers can be used under TCF:

- CA ADS application compiler (ADSA)
- CA ADS dialog compiler (ADSC)
- CA OLQ (OLQ)
- Automatic System Facility (ASF)
- IDD menu facility
- Online command facility (OCF)
- Online map compiler (MAPC)
- Online IDD
- Online schema, subschema, and system generation compilers

Transfer of control under TCF for each of these development tools is discussed in this chapter.

This section contains the following topics:

CA ADS and the Online Map Compiler

To exit from the CA ADS application or dialog compiler or the online map compiler, you can use either of these methods:

- Select the SWITCH activity from the action bar of the main menu of the CA ADS application and dialog compilers or the online map compiler.
- Use PF3 from the main menu screen.

These methods are discussed as follows.

Select the SWITCH Activity from the Main Menu

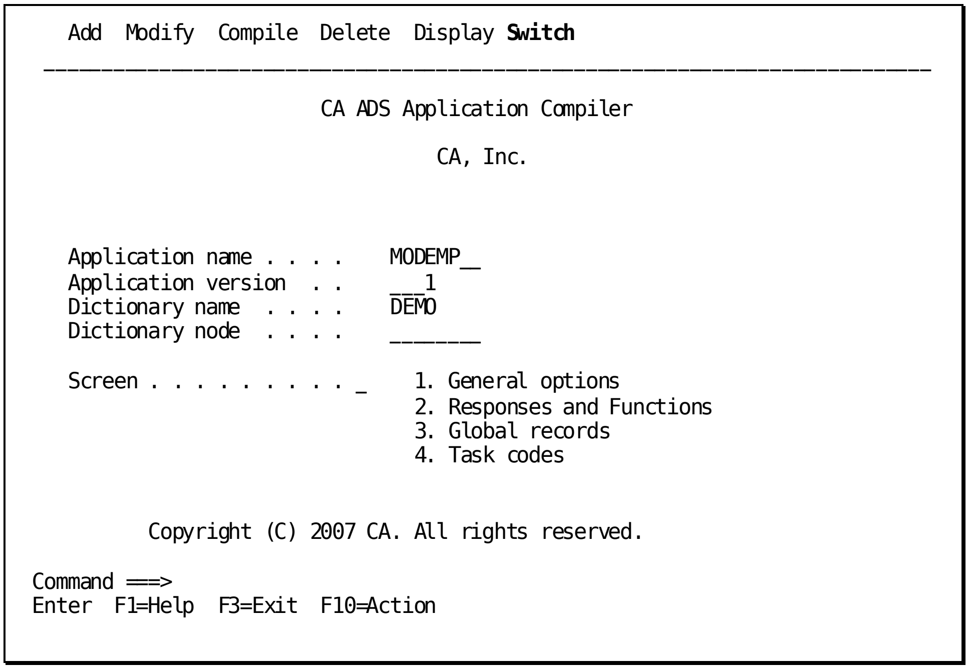
To transfer from either of the CA ADS compilers or the online map compiler to other development tools or TCF, select the SWITCH activity from the main menu. The SWITCH activity is only available from the main menu of the CA ADS compilers or the online map compiler.

When you select the SWITCH activity another menu is displayed. You can enter the task or product code for the development tool you want to transfer to from this activity. The table at the end of this section indicates where TCF transfers control depending on how you exit the CA ADS compilers or the online map compiler.

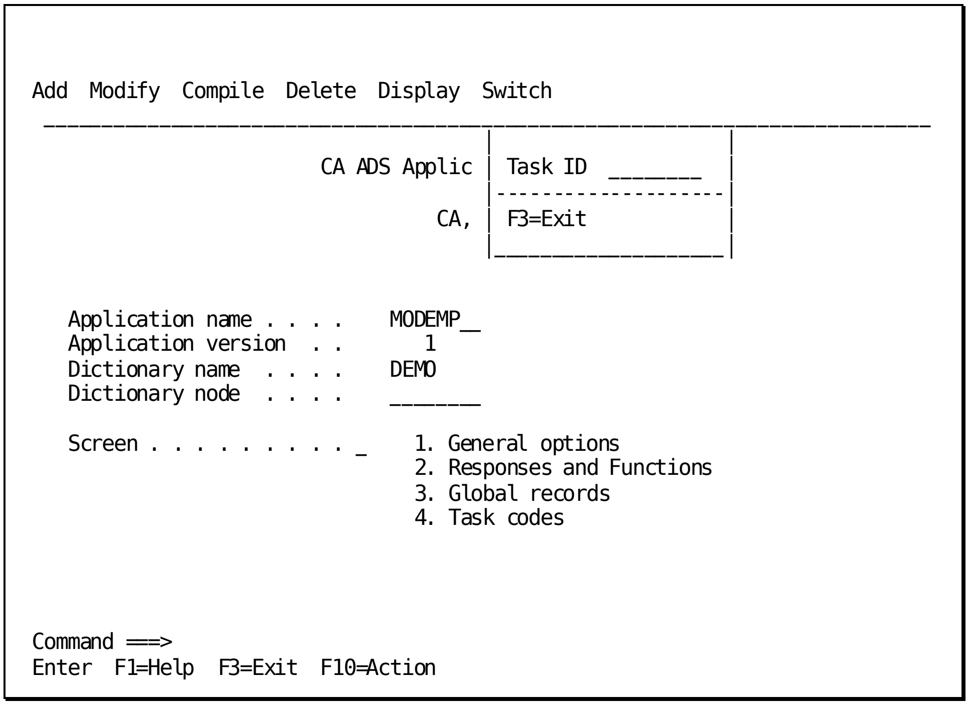
Example

The following example shows how you use the SWITCH activity of the CA ADS application compiler. Remember that the SWITCH activity works the same in both the CA ADS compilers and the online map compiler.

Using the SWITCH Activity



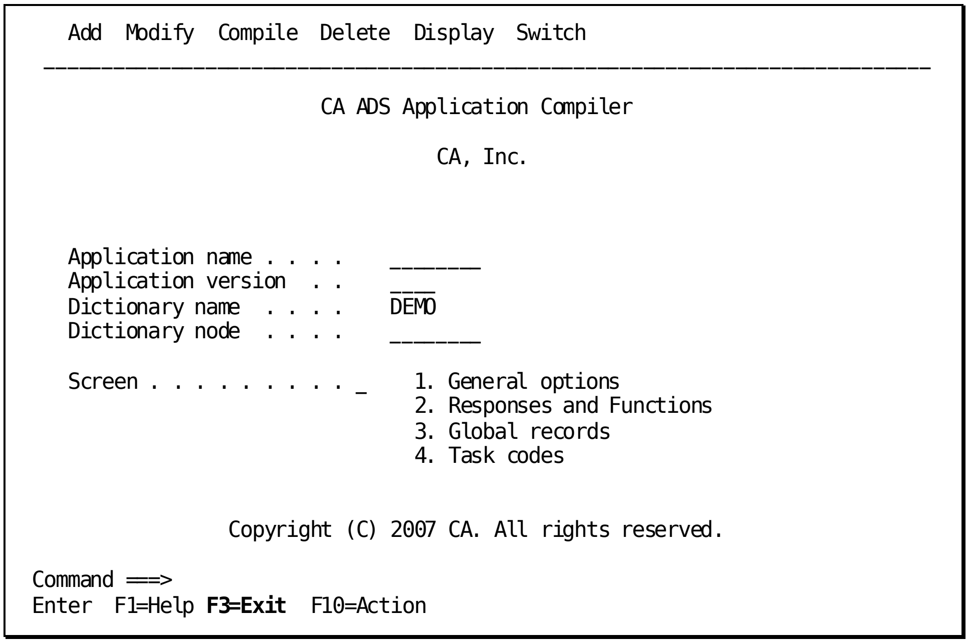
```
   Add  Modify  Compile  Delete  Display Switch
 ______________________________________________________________________________

                          CA ADS Application Compiler

                                    CA, Inc.



   Application name . . . .    MODEMP__
   Application version  . .    ___1
   Dictionary name  . . . .    DEMO
   Dictionary node  . . . .    ________

   Screen . . . . . . . . . _    1. General options
                                 2. Responses and Functions
                                 3. Global records
                                 4. Task codes


          Copyright (C) 2007 CA. All rights reserved.

Command ===>
Enter  F1=Help  F3=Exit  F10=Action
```



```
Add  Modify  Compile  Delete  Display  Switch
 ______________________________________________________________________________
                                      |                    |
                          CA ADS Applic | Task ID  ________  |
                                      |--------------------|
                                 CA,  | F3=Exit            |
                                      |____________________|


   Application name . . . .    MODEMP__
   Application version  . .       1
   Dictionary name  . . . .    DEMO
   Dictionary node  . . . .    ________

   Screen . . . . . . . . . _    1. General options
                                 2. Responses and Functions
                                 3. Global records
                                 4. Task codes




Command ===>
Enter  F1=Help  F3=Exit  F10=Action
```

Using a Program Function Key

You can exit only from the main menu screen of the CA ADS compilers and the online map compiler using PF3.



```
   Add  Modify  Compile  Delete  Display  Switch
 ------------------------------------------------------------------------------

                        CA ADS Application Compiler

                                  CA, Inc.



   Application name . . . .    ________
   Application version  . .    ____
   Dictionary name  . . . .    DEMO
   Dictionary node  . . . .    ________

   Screen . . . . . . . . . _    1. General options
                                 2. Responses and Functions
                                 3. Global records
                                 4. Task codes


                Copyright (C) 2007 CA. All rights reserved.

Command ===>
Enter  F1=Help F3=Exit  F10=Action
```

Transferring from CA ADS Compilers or the Online Map Compiler

The way your current CA ADS and map compiler sessions end and TCF transfers control on exit depends on the exit method you use. The following table summarizes what happens when you exit from either CA ADS or the map compiler under TCF.

Each suspended session is saved in a queue record; the length of time queue records are retained is specified at system generation time and can vary from site to site.

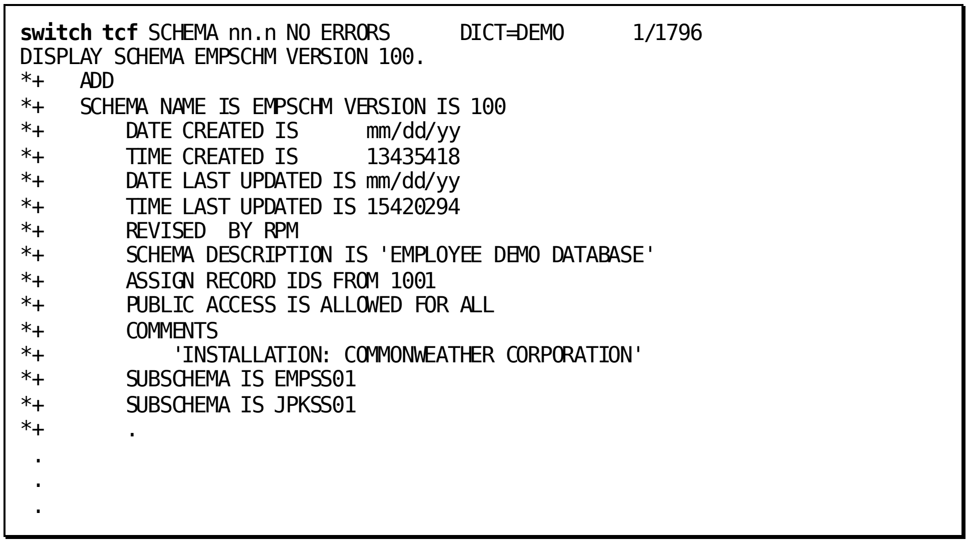
Exit Method	TCF Actions
Selecting the SWITCH activity from the action bar of the main menu	■ Suspends your session. ■ Transfers control to the requested development tool. If you do not request a tool, control transfers to: ■ The DC/UCF screen, if you do not have a suspended session ■ The development tool session that you most recently suspended, if you have a suspended session For more information about SWITCH, see SWITCH.
PF3	■ Terminates your session. ■ Transfers control to your most recently suspended development tool session, or to the DC/UCF screen if you have no suspended sessions.

CA IDMS Online Compilers

Exiting from the Online Compilers

The online schema, subschema, and system generation compilers and the online command facility are separate development tools. You can have one suspended session of each of these compilers at a given time. For example, if you have a suspended schema compiler session, you can resume but not start a new schema compiler session. You can start a new subschema or system generation compiler session.

The online schema, subschema, and system generation compilers and the online command facility use the same screen format as online IDD (the online DDDL compiler). Commands used to exit from an online compiler session can be entered in the command line of any online compiler screen. The command line is highlighted on the following sample online schema compiler screen.



```
switch tcf SCHEMA nn.n NO ERRORS      DICT=DEMO      1/1796
DISPLAY SCHEMA EMPSCHM VERSION 100.
*+   ADD
*+   SCHEMA NAME IS EMPSCHM VERSION IS 100
*+       DATE CREATED IS      mm/dd/yy
*+       TIME CREATED IS      13435418
*+       DATE LAST UPDATED IS mm/dd/yy
*+       TIME LAST UPDATED IS 15420294
*+       REVISED  BY RPM
*+       SCHEMA DESCRIPTION IS 'EMPLOYEE DEMO DATABASE'
*+       ASSIGN RECORD IDS FROM 1001
*+       PUBLIC ACCESS IS ALLOWED FOR ALL
*+       COMMENTS
*+           'INSTALLATION: COMMONWEATHER CORPORATION'
*+       SUBSCHEMA IS EMPSS01
*+       SUBSCHEMA IS JPKSS01
*+       .
 .
 .
 .
```

When you exit from an online compiler, TCF suspends or terminates your online compiler session and transfers control, as described in the following table. The contents of each suspended online compiler session are kept in a scratch record and saved until you sign off from the DC/UCF system.

Note: Do not use SUSPEND to exit the schema or subschema compilers if you are using the Automatic System Facility (ASF) in the same online session. In this case, use END or the signoff command.

Transferring from the Online Compilers

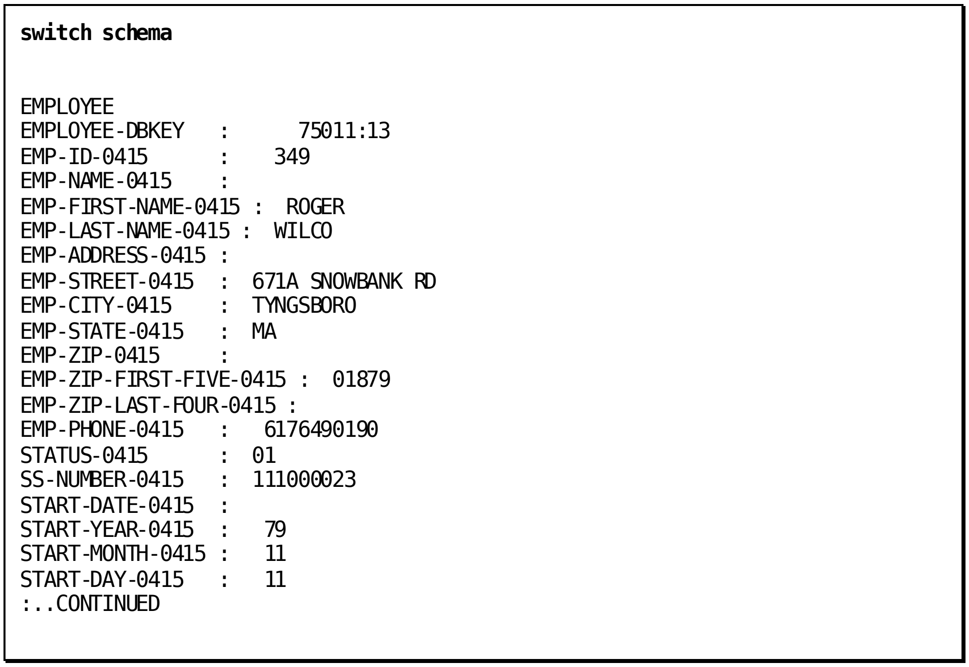
The way your current online compiler session ends and TCF transfers control on exit depends on the exit method you use, as described in the following table.

Exit Method	TCF Actions
SUSPEND command	■ Suspends your compiler session. ■ Transfers control to your most recently suspended development tool session or to the Selection screen if you have no sessions.
SWITCH command	■ Suspends your compiler session. ■ Transfers control to the requested development tool. If you do not request a tool, control transfers as if you used SUSPEND (see the description for SUSPEND command). For more information about SWITCH, see SWITCH.
BYE, QUIT, or SIGNOFF command (or associated function key)	■ Terminates your compiler session. ■ Transfers control to your most recently suspended development tool session or to the Selection screen if you have no suspended sessions.

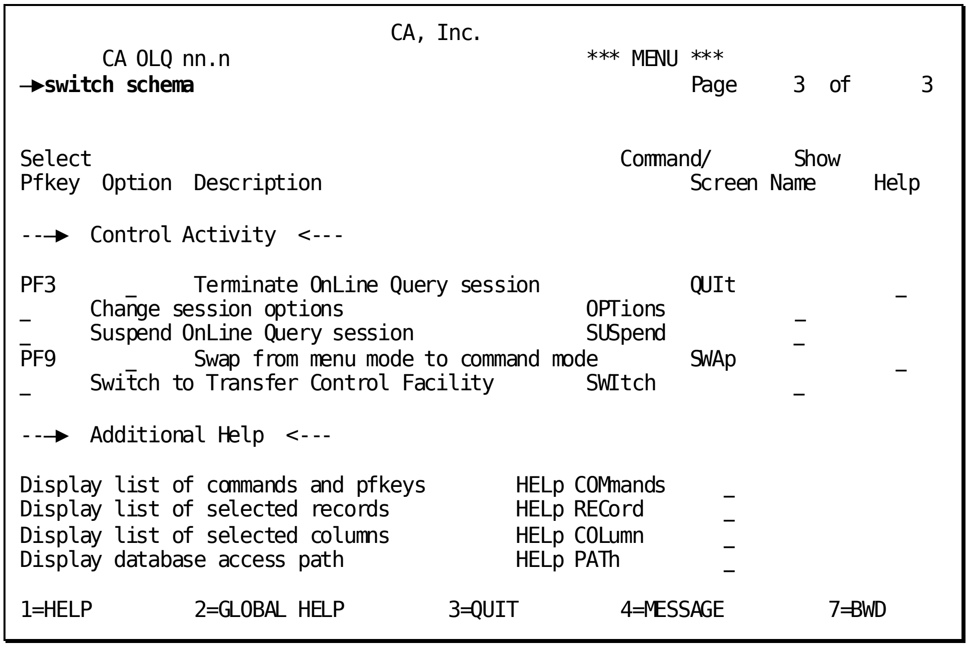
CA OLQ

Exiting from CA OLQ

Commands that exit you from CA OLQ can be specified in the command line on CA OLQ screens. The command line is highlighted on each of the following sample CA OLQ screens.



```
switch schema


EMPLOYEE
EMPLOYEE-DBKEY     :      75011:13
EMP-ID-0415        :    349
EMP-NAME-0415      :
EMP-FIRST-NAME-0415 :  ROGER
EMP-LAST-NAME-0415 :  WILCO
EMP-ADDRESS-0415 :
EMP-STREET-0415    :  671A SNOWBANK RD
EMP-CITY-0415      :  TYNGSBORO
EMP-STATE-0415     :  MA
EMP-ZIP-0415       :
EMP-ZIP-FIRST-FIVE-0415 :  01879
EMP-ZIP-LAST-FOUR-0415 :
EMP-PHONE-0415     :   6176490190
STATUS-0415        :  01
SS-NUMBER-0415     :  111000023
START-DATE-0415    :
START-YEAR-0415    :   79
START-MONTH-0415 :   11
START-DAY-0415     :   11
:..CONTINUED
```



```
                                CA, Inc.
       CA OLQ nn.n                                 *** MENU ***
-►switch schema                                        Page    3  of     3


Select                                             Command/      Show
Pfkey  Option  Description                            Screen Name     Help

--► Control Activity  <---

PF3      _      Terminate OnLine Query session        QUIt              _
_     Change session options                      OPTions       _
_     Suspend OnLine Query session                SUSpend       _
PF9      _      Swap from menu mode to command mode   SWAp              _
_     Switch to Transfer Control Facility         SWItch        _

--► Additional Help  <---

Display list of commands and pfkeys       HELp COMmands    _
Display list of selected records          HELp RECord      _
Display list of selected columns          HELp COLumn      _
Display database access path              HELp PATh        _

1=HELP        2=GLOBAL HELP        3=QUIT        4=MESSAGE        7=BWD
```

Function keys can also be used to exit from CA OLQ. Function keys for CA OLQ are defined at system generation time and can vary from site to site.

When you exit from CA OLQ, TCF suspends or terminates your CA OLQ session and transfers control as described in the following table. The contents of a suspended CA OLQ session are kept in a scratch record and saved until you sign off from the DC/UCF system.

Note: You can have a maximum of one suspended CA OLQ session. For example, you can resume but not start a new CA OLQ session if you already have a suspended CA OLQ command mode or menu facility session.

Transferring from CA OLQ

The way your current CA OLQ session ends and TCF transfers control on exit depends on the exit method you use, as described in the following table.

Exit Method	TCF Actions
SUSPEND command (command mode only)	■ Suspends your CA OLQ session(1). ■ Transfers control to your most recently suspended development tool session or to the Selection screen if you have no sessions.

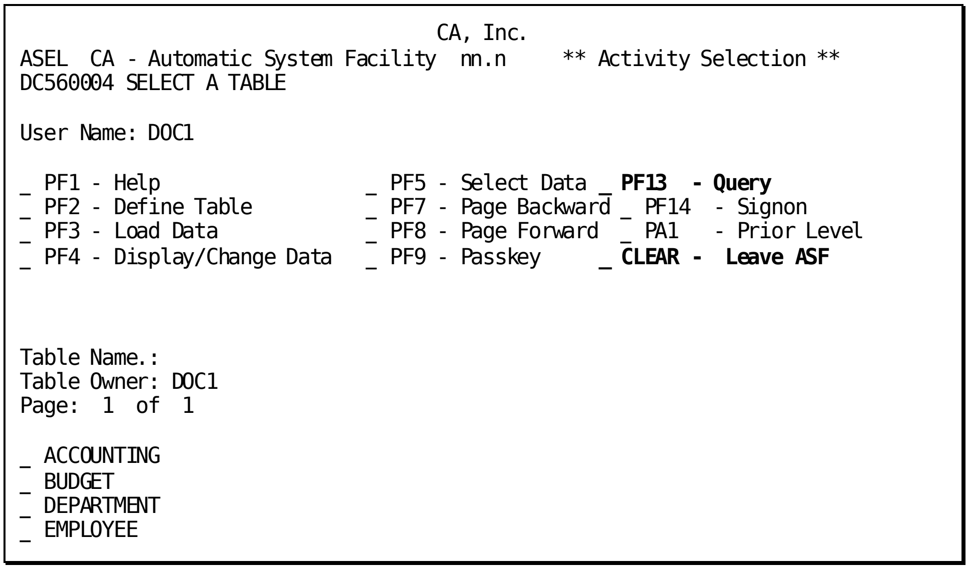
Exit Method	TCF Actions
SWITCH command	■ Suspends your CA OLQ session. ■ Transfers control to the requested development tool. If you do not request a tool, control transfers as if you used SUSPEND (see above). For more information about SWITCH, see SWITCH.
BYE, QUIT, or SIGNOFF command (or associated function key)	■ Terminates your CA OLQ session. ■ Transfers control to your most recently suspended development tool session or to the Selection screen if you have no suspended sessions.

(1) An identifier for the suspended CA OLQ session is not added to your list of suspended sessions.

ASF

Exiting from ASF

Fields highlighted on the following sample ASF screen can be used to exit from ASF.



```
                                  CA, Inc.
ASEL  CA - Automatic System Facility  nn.n     ** Activity Selection **
DC560004 SELECT A TABLE

User Name: DOC1

_ PF1 - Help                   _ PF5 - Select Data _ PF13  - Query
_ PF2 - Define Table           _ PF7 - Page Backward _ PF14  - Signon
_ PF3 - Load Data              _ PF8 - Page Forward  _ PA1   - Prior Level
_ PF4 - Display/Change Data    _ PF9 - Passkey      _ CLEAR -  Leave ASF



Table Name.:
Table Owner: DOC1
Page:  1  of  1

_ ACCOUNTING
_ BUDGET
_ DEPARTMENT
_ EMPLOYEE
```

Function keys can also be used to exit ASF or to switch to CA OLQ. To leave ASF entirely, use CLEAR. To switch to CA OLQ, use PF13 from the Activity Selection screen or type OLQ in the simulated PF key field of any screen.

When you exit from ASF, TCF maintains the ASF session and transfers control as described in the following table. However, an ASF session is not suspended in the usual sense when you leave ASF by transferring control to CA OLQ. Instead, TCF maintains your ASF signon information and retains currency on the table, if one was selected. When you select ASF from your suspended sessions list, TCF uses the retained signon and table information, and displays the ASF Activity Selection screen regardless of the screen you were using when you transferred control to CA OLQ.

Transferring from ASF

The way your current ASF session ends and TCF transfers control on exit depends on the exit method you use, as described in the following table.

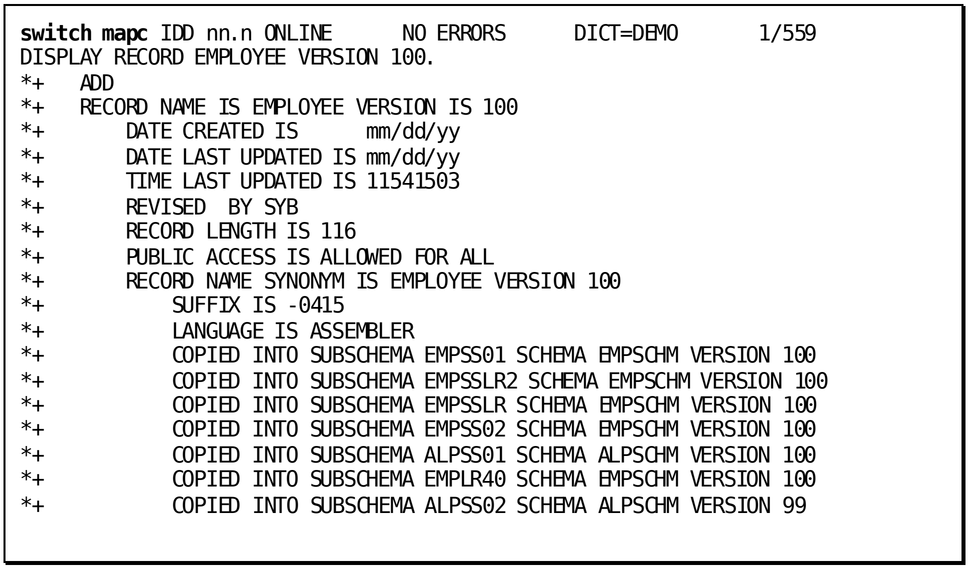
Exit Method	TCF Actions
PF13--Query	■ Transfers control to CA OLQ, saving signon and table information.

Exit Method	TCF Actions
CLEAR--Leave ASF	■ Terminates your ASF session. ■ Transfers control to your most recent suspended development tool session, or to the DC/UCF screen if you have no suspended sessions.

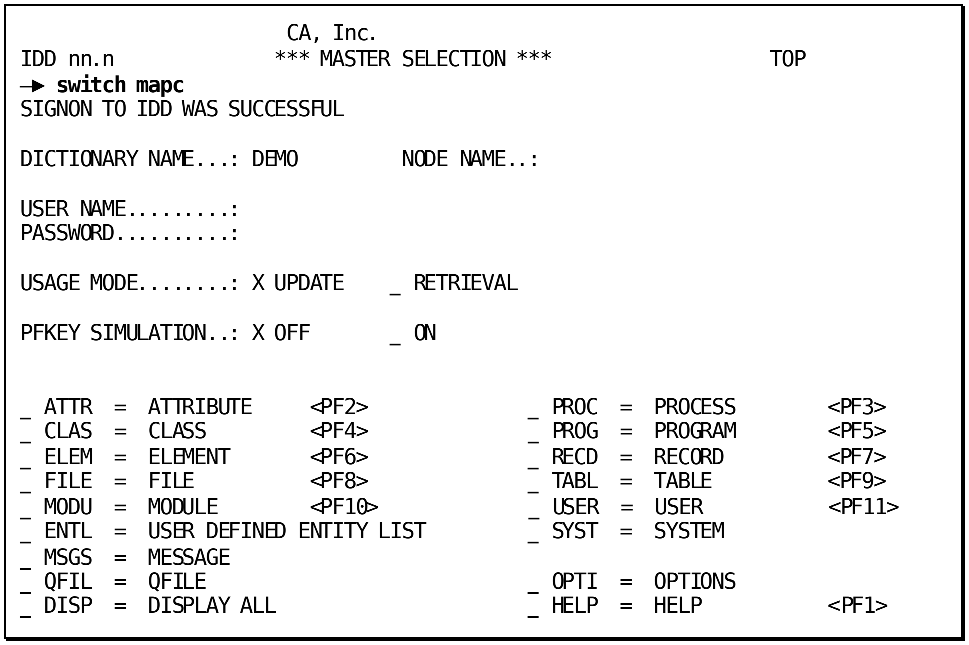
IDD

Exiting from IDD

Commands used to exit from IDD can be specified on the command line on IDD screens. The command line is highlighted on each of the following sample IDD screens.



```
switch mapc IDD nn.n ONLINE      NO ERRORS      DICT=DEMO       1/559
DISPLAY RECORD EMPLOYEE VERSION 100.
*+   ADD
*+   RECORD NAME IS EMPLOYEE VERSION IS 100
*+       DATE CREATED IS      mm/dd/yy
*+       DATE LAST UPDATED IS mm/dd/yy
*+       TIME LAST UPDATED IS 11541503
*+       REVISED  BY SYB
*+       RECORD LENGTH IS 116
*+       PUBLIC ACCESS IS ALLOWED FOR ALL
*+       RECORD NAME SYNONYM IS EMPLOYEE VERSION 100
*+           SUFFIX IS -0415
*+           LANGUAGE IS ASSEMBLER
*+           COPIED INTO SUBSCHEMA EMPSS01 SCHEMA EMPSCHM VERSION 100
*+           COPIED INTO SUBSCHEMA EMPSSLR2 SCHEMA EMPSCHM VERSION 100
*+           COPIED INTO SUBSCHEMA EMPSSLR SCHEMA EMPSCHM VERSION 100
*+           COPIED INTO SUBSCHEMA EMPSS02 SCHEMA EMPSCHM VERSION 100
*+           COPIED INTO SUBSCHEMA ALPSS01 SCHEMA ALPSCHM VERSION 100
*+           COPIED INTO SUBSCHEMA EMPLR40 SCHEMA EMPSCHM VERSION 100
*+           COPIED INTO SUBSCHEMA ALPSS02 SCHEMA ALPSCHM VERSION 99
```



```
                       CA, Inc.
IDD nn.n              *** MASTER SELECTION ***                  TOP
→ switch mapc
SIGNON TO IDD WAS SUCCESSFUL

DICTIONARY NAME...: DEMO         NODE NAME..:

USER NAME.........:
PASSWORD..........:

USAGE MODE........: X UPDATE    _ RETRIEVAL

PFKEY SIMULATION..: X OFF       _ ON


_ ATTR  =  ATTRIBUTE     <PF2>              _ PROC  =  PROCESS       <PF3>
_ CLAS  =  CLASS         <PF4>              _ PROG  =  PROGRAM       <PF5>
_ ELEM  =  ELEMENT       <PF6>              _ RECD  =  RECORD        <PF7>
_ FILE  =  FILE          <PF8>              _ TABL  =  TABLE         <PF9>
_ MODU  =  MODULE        <PF10>             _ USER  =  USER          <PF11>
_ ENTL  =  USER DEFINED ENTITY LIST         _ SYST  =  SYSTEM
_ MSGS  =  MESSAGE
_ QFIL  =  QFILE                            _ OPTI  =  OPTIONS
_ DISP  =  DISPLAY ALL                      _ HELP  =  HELP          <PF1>
```

When you exit from IDD, TCF suspends or terminates your IDD session and transfers control as described in the following table. The contents of a suspended IDD session are kept in a scratch record and are retained until you sign off of the DC/UCF system.

Note: You can have a maximum of one suspended online IDD session and one suspended IDD menu facility session. For example, if you have a suspended online IDD session, you can start a new IDD menu facility session but cannot start another new online IDD session. This is because an online IDD session is totally unrelated to a simultaneous IDD menu facility session.

Transferring from IDD

The way your current IDD session ends and TCF transfers control on exit depends on the exit method you use, as described in the following table.

Exit Method	TCF Actions
SUSPEND command	■ Suspends your IDD session. ■ Transfers control to your most recently suspended development tool session, or to the Selection screen if you have no suspended sessions.

Exit Method	TCF Actions
SWITCH command	■ Suspends your IDD session. ■ Transfers control to the requested development tool. If you do not request a tool, control transfers as if you used SUSPEND For more information about SWITCH, see SWITCH.
BYE, END, LOGOFF, or SIGNOFF command or CLEAR key, as appropriate	■ Terminates your IDD session. ■ Transfers control to your most recently suspended development tool session, or to the Selection screen if you have no suspended sessions.

Chapter 13: SYSIDMS Parameter File

A SYSIDMS parameter is a parameter that can be added to the JCL stream of a batch job running in local mode or under the central version.

This section contains the following topics:

SYSIDMS Overview

You can use SYSIDMS parameters to specify the following:

- Physical requirements of the environment, such as the DMCL and database to use at runtime
- Runtime directives that assist in application execution
- Operating system-dependent file information

For a complete list of the parameters that can be specified, see Parameter Descriptions.

Establishing Site Defaults

Site-specific defaults can be established for all SYSIDMS parameters by assembling a SYSIDMS defaults load module. If it exists, this load module is used at runtime to determine the default values for all SYSIDMS parameters. Defaults may then be overridden in an individual job step by including a SYSIDMS parameter file in the execution JCL.

Creating a SYSIDMS Defaults Load Module

The following example illustrates how to code a SYSIDMS defaults load module. It is a table of 80-character constants, each of which may contain one or more SYSIDMS parameters, as described in Parameter Descriptions. A parameter and its value must be contained within a single 80-character constant, but more than one parameter may appear within a constant. The last constant must have a value of "END SYSIDMS DEFAULTS."

```
TITLE 'SYSIDMS - Build load module for SYSIDMS defaults'
SYSIDMS START 0
**************************************************************
* Code any SYSIDMS parms that you want to be part of this SYSIDMS
* defaults load module.  This SYSIDMS defaults load module will be
* processed first before trying to process any SYSIDMS parms defined
* in the JCL for any IDMS batch job.
***************************************************************
          SPACE
          DC  CL80'ECHO=ON DMCL=GLBLDMCL'
          DC  CL80'JOURNAL=OFF'
          SPACE
* The following statement is mandatory and must be the last statement
* in the SYSIDMS defaults load module.
          DC  CL80'END SYSIDMS DEFAULTS'
          END
```

Linking a SYSIDMS Defaults Load Module

The load module must have both a name and an entry point of SYSIDMS. For operating systems that support XA storage, the load module can be linked as AMODE 31, RMODE ANY.

Overriding SYSIDMS Parameter Defaults

SYSIDMS default values can be overridden for an individual job step by including a SYSIDMS parameter file in the execution JCL.

In the following example, the SYSIDMS parameters included in the job stream instruct CA IDMS/DB to use the DMCL LOCLDMCL to execute a job. DBNAME identifies EMPDB as the database to access at runtime, and the QSAM parameters instruct CA IDMS/DB to use the IDMSQSAM look-ahead read facility when accessing EMPSEG.EMPAREA:

```
//SYSIDMS  DD *
DMCL=LOCLDMCL DBNAME=EMPDB
IDMSQSAM=ON  QSAMAREA=EMPSEG.EMPAREA
```

In the following example, the SYSIDMS parameters used are typical for a batch job running under the central version:

```
//SYSIDMS  DD *
DBNAME=EMPDB  NODENAME=SYSTEM90
```

Parameter Summary

Debugging and Abend Control Parameters

Parameter	CV	Batch	z/OS	z/VSE	z/VM
ABEND_ON_DEADLOCK	X		X	X	X
ABEND_SVC_DUMP	X		X		
ABENDTRACE		(1)			
ABENDTRACE_ENTRIES		(1)			
ABENDTRACE_ SUBSCHEMA_DISPLAY		(1)			
ABENDTRACE_VIBSNAP		(1)			
ADJUNCT_TRACE_TABLE	X	X	X	X	X
AREA_VALIDATION_ MSGS	X		X	X	X
DB_DEADLOCK_DUMP	X		X	X	X
DC_DEADLOCK_DUMP	X		X	X	X
DC_DEADLOCK_0029	X		X	X	X
DEADLOCK_ABEND_ERUS	X		X	X	X
DEADLOCK_ABEND_0029	X		X	X	X

Parameter	CV	Batch	z/OS	z/VSE	z/VM
DEADLOCK_DETAILS	X		X	X	X
DMLTRACE		X	X	X	X
ECHO	X	X	X	X	X
PROCTRACE		X	X	X	X
QSAMTRACE		X	X	X	X
SQLTRACE		X	X	X	X
SYSTRACE	X		X	X	X
TRACE_TABLE_SIZE	X		X	X	X

(1) You must have CA Optimizer/II or CA SymDump Batch installed to use this parameter.

Performance-Related Parameters

Parameter	CV	Batch	z/OS	z/VSE	z/VM
BUFFER_PURGE		X	X	X	X
BUFFERSTAT		X	X	X	X
DBIO_HICCUP	X		X	X	X
DLBLMOD		X		X	
FILE_BUF		X	X	X	X
IDMSQSAM		X	X	X	X
LIMIT_PREFETCH		X	X	X	X
PREFETCH	X	X	X	X	X
PREFETCH_BUF		X	X	X	X
QSAMAREA		X	X	X	X
QSAMBUF#		X	X		X
QSAM#BUF		X			
QSAMTRACE		X	X	X	X
SCRATCH_IN_STORAGE		X	X	X	X
SQL_CACHE_ENTRIES		X	X	X	X
SQL_INTLSORT		X	X	X	X
TRANSACTION_SHARING		X	X	X	X

File-Related Parameters

Parameter	CV	Batch	z/OS	z/VSE	z/VM
DLBLMOD		X		X	
DYNALLOC_WAIT	(2)	X	X		
DYNALLOC_WAIT_SECONDS	(2)	X	X		
FILE_BUF		X	X	X	X
LENGTH_PAGE		X	X	X	X
LIST		X			
LOCAL_DYNAMIC_ ALLOCATION		X	X	X	X
MULTIDSN		X		X	
OVERPRINT		X			
ROLLBACK3490		X	X	X	X
SYS_MSG		X	X	X	X
SYSTRK_DDNAME_PREFIX	X	X	X	X	X
UPPER		X	X	X	X
WIDTH_PAGE		X	X	X	X

(2) Applies to CV startup only.

z/VSE File-Related Parameters

Parameter	CV	Batch	z/OS	z/VSE	z/VM
FILENAME		X		X	
BLKSIZE		X		X	
BLOCKS		X		X	
DEVADDR		X		X	
FILABL		X		X	
FILETYPE		X		X	
LRECL		X		X	
RECFM		X		X	
REWIND		X		X	

Note: For more information on these parameters, see the dedicated z/VSE File Parameters (see page 169) section.

Connection and Environment Parameters

Parameter	CV	Batch	z/OS	z/VSE	z/VM
CICS_NAME		X			
CVMACH		X			X
CVNUM		X			X
CVRETRY		X	X	X	X
CVRETRY_MSG_CODES		X	X	X	X
DBNAME		X	X	X	X
DICTNAME		X	X	X	X
DICTNODE		X	X	X	X
DMCL	X	X	X	X	X
ENABLE_RRS		X	X		
LANG		X	X	X	X
LOCAL		X	X	X	X
NODENAME		X	X	X	X
REREAD_SYSCTL		X	X	X	X
SYSCTL		X	X	X	X

Miscellaneous Runtime Directives

Parameter	CV	Batch	z/OS	z/VSE	z/VM
ARCHIVE_JOURNAL_WARNING_PERCENT		X	X	X	X
BCF_INPUT_80		X	X	X	X
IGNORE_SYSTRK_DMCL	X		X	X	X
CV_STARTUP_XA_REGION_MB	X		X	X	X
DBAN_SORT_PASSES		X	X	X	X
DC_SCRATCH		X	X	X	X
DCNAME	X		X		

Parameter	CV	Batch	z/OS	z/VSE	z/VM
DISABLE_SVC_SCREEN	X		X	X	X
DMCL	X	X	X	X	X
DSGROUP	X		X		
EVAL_BASE_YEAR	X	X	X	X	X
EVAL_CENTURY_VALIDATION	X	X	X	X	X
EVAL_LOW_CENTURY	X	X	X	X	X
EVAL_HIGH_CENTURY	X	X	X	X	X
JOURNAL		X	X	X	X
JRNLDTS	X		X	X	X
LOADAREA		X	X	X	X
LOCAL_NOJOURNAL_ RETRIEVAL		X	X	X	X
LOCALPUR		X	X	X	X
MASTERKEY_EXITS	X		X	X	X
MSGDICT		X	X	X	X
PARM		X	X	X	X
SORT FIELD MAX LEN		X	X	X	X
SORTSIZE		X	X	X	X
TASK_ANALYZER_EXITS	X		X	X	X
PDAT_LOCATION	X		X	X	
WORK		X		X	
DATE_SIMULATOR_SVC	X	X	X		
SHUTDOWN_STALL_TIME	X		X	X	X
PRMPT_MSG	X		X	X	X
SAVE_SQL_SYNTAX	X		X	X	X

TCP/IP-Related Parameters

Parameter	CV	Batch	z/OS	z/VSE	z/VM
TCP/IP_STATUS	X		X	X	X
INCLUDE_TCP/IP_STACK	X		X		X
EXCLUDE_TCP/IP_STACK	X		X		

Parameter	CV	Batch	z/OS	z/VSE	z/VM
TCP/IP_STACK_1	X				X
TCP/IP_STACK_2	X				X
TCP/IP_STACK_3	X				X
TCP/IP_STACK_4	X				X
TCP/IP_STACK_5	X				X
TCP/IP_STACK_6	X				X
TCP/IP_STACK_7	X				X
TCP/IP_STACK_8	X				X

Parameter Descriptions

ABEND_ON_DEADLOCK

Forces the abnormal termination of a task that encounters a database resource deadlock. In normal CV operations, a database resource deadlock results in control being returned to the application program with an indication that a deadlock occurred. This parameter causes the task to be abended instead.

Note: It is meaningful only in the SYSIDMS file associated with a central version.

ABEND_SVC_DUMP=ON|OFF

Causes an SVC dump to be requested when CV abends prematurely (ON) or not at all (OFF).

Default: ON

Note: Specifying OFF is not recommended. If the CV address space abends, a dump will be required to determine the cause of the abend.

ABENDTRACE=ON|OFF

Activates the tracing of various pieces of CA IDMS data when using CA Optimizer/II or CA SymDump Batch. This parameter is meaningful only in the SYSIDMS file associated with a batch job.

Note: You must have CA Optimizer/II or CA SymDump Batch installed to use this parameter. However, if CA Optimizer/II or CA SymDump Batch is installed and you do not specify ABENDTRACE=ON, you can still get an abbreviated trace of CA IDMS activity by setting optional bit 265 in RHDCOPTF.

ABENDTRACE_ENTRIES=*nnn*

Overrides the default number of entries being traced by ABENDTRACE. This parameter is meaningful only in the SYSIDMS file associated with a batch job.

Note: You must have CA Optimizer/II or CA SymDump Batch installed to use this parameter.

ABENDTRACE_SUBSCHEMA_DISPLAY=ON

Activates the display of information from the subschema in use at the time of abend when using ABENDTRACE. This parameter is meaningful only in the SYSIDMS file associated with a batch job.

Note: You must have CA Optimizer/II or CA SymDump Batch installed to use this parameter.

ABENDTRACE_VIBSNAP=ON

Causes the dump of the VIB at the time of abend when using ABENDTRACE. This parameter is meaningful only in the SYSIDMS file associated with a batch job.

Note: You must have CA Optimizer/II or CA SymDump Batch installed to use this parameter.

ADJUNCT_TRACE_TABLE=table-size KB|MB

Specifies the size of the adjunct trace table in kilobytes (KB) or megabytes (MB).

Limit: 0–9999

Default: 0

Note: Any non-zero adjunct table size established by a SYSIDMS parameter overrides the adjunct trace table size specified in the system definition.

ARCHIVE_JOURNAL_WARNING_PERCENT=*percent-number*

Specifies the threshold percent that ARCHIVE JOURNAL uses to issue a warning message that a journal file is nearly full. If the amount of available space after condensing a journal is less than percent-number, ARCHIVE JOURNAL issues a warning message indicating that the journal is nearly full of condensed segments.

Default: 10 percent

AREA_VALIDATION_MSGS=ON|OFF

ON causes the informational messages DB347042 and DB347043 to be displayed on the JES log during startup and shutdown for each area being shared in a SYSPLEX data sharing environment. If you are sharing many areas this can cause the JES log to be congested.

Default: OFF

Note: This parameter is only applicable in a SYSPLEX data sharing environment.

BCF_INPUT_80=ON|OFF

IDMSBCF input is done over 80 characters by a #LINEIN INAREA=CARD,MAXIN=80,... statement. However, afterwards all information starting from column 73 until column 80 is cleared and overwritten by spaces. That is, the actual maximum input size is only 72 characters.

Executing any SQL PUNCH command may produce a SYSPCH output exceeding column 72. When using this output as input for an IDMSBCF job afterwards, may result in possible syntax errors because the input is truncated after column 72.

By specifying SYSIDMS parameter BCF_INPUT_80=ON, information between columns 73 and 80 is not cleared (that is, input for BCF will be over 80 characters). In all other cases (that is, no BCF_INPUT_80 parameter or not specifying BCF_INPUT_80=OFF), it defaults to 72 characters.

Default: OFF

BUFFER_PURGE

Causes updated pages to be written to the database whenever the number of buffers containing such pages exceeds 1/4 of the number of pages in the buffer pool. This parameter may improve the performance of local mode update jobs that do not issue frequent COMMITs, since it will make buffers available for the use of pre-fetch. It has meaning only for local mode batch jobs.

BUFFERSTAT

Produces a report containing buffer pool I/O statistics that can be used for tuning purposes. The report will be written to SYSLST at the end of the job. This parameter has meaning only for local mode batch jobs.

Note: For a description of the fields in the report, see the BUFFERSTAT Report Field Descriptions (see page 166).

CICS_NAME=*cics-name*

Specifies a 1 to 4 character value that identifies the CICS system name being started.

CVMACH=*cms-machine-name*

(z/VM only) Specifies the virtual machine in which the DC/UCF system is executing.

CVNUM=*nnn*

(z/VM only) Specifies the number of the central version that is accessible by CMS and is used to route database requests through the IDMSVMCF facility.

Limits: 0-255

CVRETRY=ON|OFF

ON indicates that the following message is displayed on the operator console when the CA IDMS central version is not active:

```
CV nnn NOT ACTIVE. REPLY RETRY OR CANCEL.
```

Default: ON

CVRETRY_MSG_CODES=descriptor-route-codes

Specifies the descriptor and route codes to be used for batch message DC208002 (CV cv-number NOT ACTIVE. REPLY RETRY OR CANCEL). descriptor-route-codes must be an eight-digit hexadecimal value.

The first four digits of descriptor-route-codes represent the descriptor codes and the last four digits represent the route codes. Each bit within the descriptor or route codes represents a code value. The first bit (x’8000’) represents code value 1 and the last bit (x’0001’) represents code value 16. Multiple bits can be on in each set of codes so that x’8101’ represents code values 1, 8 and 16.

Default: 00004000 (representing descriptor code zero (0) and route code two)

Note: descriptor-route-codes value is automatically considered to be in the hexadecimal format.

CV_STARTUP_XA_REGION_MB=*nnn*

Specifies the size of the initial XA storage pool acquired during early CV startup.

nnn

Specifies the size in MB (megabytes) of the initial XA storage pool.

Default: 32 MB

DB_DEADLOCK_DUMP

Specifies that a dump will be produced for a task that is abended due to a database resource deadlock. This parameter is used in conjunction with the ABEND_ON_DEADLOCK parameter. If not specified, no dump will be produced when a task is abended due to a database deadlock.

DBIO_HICCUP=nn

By default, the CV will generate a hiccup wait if a task accesses 100 pages without generating a wait. This parameter allows 1-99 pages to be accessed before a wait is generated. Specifying a value greater than 100 results in a page count of 100.

Note: It is recommended that you do not use very low values.

DBAN_SORT_PASSES=*sort-passes*

Specifies the number of sorts that IDMSDBAN uses when auditing index orphan chains.

Default: 1

The higher the number of sorts, the greater the number of errors that can be detected, if any exist. However, increasing the number of sorts also increases the overhead for auditing orphan chains, which can be significant when processing large indexes. Because most errors are detected with a single sort, the default of one is sufficient in most cases. You should increase the number of sorts beyond 1 only when you need to identify every error in an index.

DBNAME=*database-name*

For non-SQL applications, specifies the name of the database to access at runtime. database-name is either a segment name or a DBNAME defined in a database name table. For SQL applications, it has no impact.

DC_DEADLOCK_NODUMP

Specifies that a dump is not produced for a task that is abended due to a DC resource deadlock. This parameter overrides the DUMP/NODUMP sysgen parameter.

DC_DEADLOCK_0029

Specifies that tasks that encounter a DC resource deadlock be abended with a code of 0029 rather than a code of DEAD.

DC_SCRATCH=ON|OFF

ON allows local jobs to use the central version's scratch area (DDLDCSCR) when a local scratch area (DDLOCSCR) is not defined in the DMCL.

Default: ON

DCNAME=*member-name*

(z/OS only) Specifies the member name of the system within a data sharing group. This name also becomes the system (node) name, overriding the value specified in the system definition. member-name must consist of characters A-Z, 0-9, $, #, or @.

Limits: 1-8 characters

Note: This parameter is only applicable in a data sharing environment.

DEADLOCK_ABEND_ERUS

Specifies that ERUS tasks that encounter a database resource deadlock be abnormally terminated. This parameter is meaningful only if the ABEND_ON_DEADLOCK parameter is also specified.

DEADLOCK_ABEND_0029

Specifies that tasks that are abended due to database resource deadlocks use a code of 0029 rather xx29, where "xx" represents the major code of the DML request that was being issued at the time of the abend. This parameter has meaning only if the ABEND_ON_DEADLOCK parameter is also specified.

DEADLOCK_DETAILS=ON|OFF

ON specifies that more detail be provided in a deadlock situation.

Default: OFF

DICTNAME=*dictionary-name*

Specifies a dictionary to use when loading a subschema from a load area. For dictionary-related tools like CA IDMS compilers and pre-compilers, IDMSBCF, and so on, dictionary-name specifies the dictionary to access at run time. For SQL applications, dictionary-name specifies the name of the dictionary to connect to at run time.

DICTNODE=*dictionary-node-name*

For SQL applications and dictionary-related tools under the central version, specifies the name of the DC/UCF system that controls the dictionary to access at run time. For applications running in local mode, this parameter is not applicable.

DISABLE_SVC_SCREEN=SVC-number

Specifies the number of an SVC for which screening is to be disabled.

Limits: 0–255

Disabling screening of an SVC allows it to be issued by a program executing within the DC/UCF address space.

The DISABLE_SVC_SCREEN parameter can be specified multiple times.

Note: Use this parameter with caution, since issuing unscreened SVCs within the DC/UCF system can degrade performance and result in abends. For more information about SVC screening, see DML Reference Guide for Assembler.

DLBLMOD=ON|OFF

(z/VSE only) ON specifies that the DLBL type in the disk label will be changed from 'DA' to 'SD' when sequential processing (IDMSQSAM) is activated. After the disk labels are processed as 'SD' during the QSAM file OPEN, the DLBLs are changed back to 'DA' to allow random BDAM processing.

Default: OFF

DMCL=*dmcl-name*

Specifies the name of the DMCL load module to use in local mode.

Default: IDMSDMCL

DMLTRACE=ON|OFF

ON activates a trace facility that traces all navigational DML requests made by an application.

Default: OFF

DSGROUP=*data-sharing-group-name*

(z/OS only) Specifies the name of the data sharing group of which this system is a member. All CA IDMS systems that are members of the same group must specify the same group name. The data-sharing-group-name must consist of characters A-Z, 0-9, $, #, or @. Names that begin with SYS or UNDESIG are reserved and cannot be used. Names that begin with A-I may be in use by the operating system and should be avoided.

Limits: 1-8 characters

Note: This parameter is only applicable in a data sharing environment.

DYNALLOC_WAIT=ON|OFF

ON specifies that dynamic allocation will do an ENQ wait for the DSN until it becomes available, when a data set is in use.

OFF specifies that when DYNALLOC_WAIT_SECONDS option is zero or not specified, the dynamic allocation request will fail.

This option applies to local mode jobs and to CV startup.

Default: OFF

DYNALLOC_WAIT_SECONDS=nnn

Specifies that when a data set is in use, dynamic allocation will wait for a specified interval and retry the allocation. If the DSN is still unavailable it will repeat the process until the data set is available. If this option is specified with a non-zero value, it will override the DYNALLOC_WAIT option. If zero is specified and the DYNALLOC_WAIT option is OFF or not specified; the dynamic allocation request will fail.

This option applies to local mode jobs and to CV startup.

Limits: 255

ECHO=ON|OFF

Indicates whether SYSIDMS parameters are displayed on the JES log.

Default: OFF

ENABLE_RRS=ON|OFF

(z/OS only) ON activates RRS support for batch applications.

Default: OFF

EVAL_BASE_YEAR=base-year

Specifies the base year to be assumed by the DATEDIFF and DATEOFF built-in functions. The base year is used to determine whether a two-digit year is considered to be in the twentieth or twenty-first centuries. A year whose value is greater than the base year is considered to be in the twentieth century; values less than or equal to the base year are considered to be in the twenty-first century.

Limits: 1 - 99

Default: 68

EVAL_CENTURY_VALIDATION=ON|OFF

Specifies whether century values are to be validated by built-in functions that accept four-digit years such as GOODDATEX.

ON

Validate century values.

OFF

Does not validate century values.

Default: OFF

EVAL_LOW_CENTURY=low-century

Specifies the lowest century value to be considered valid. The value specified is used during century validation in built-in functions that accept four-digit years such as GOODDATEX. Centuries are validated only if the EVAL CENTURY VALIDATION is ON.

Limits: 1 - 99

Default: 19

EVAL_HIGH_CENTURY=high-century

Specifies the highest century value to be considered valid. The value specified is used during century validation in built-in functions that accept 4-digit years such as GOODDATEX. Centuries are validated only if the EVAL CENTURY VALIDATION is ON.

Limits: 1 - 99

Default: 20

EXCLUDE_TCP/IP_STACK=stack=name

(z/OS and z/VM only) Creates an EXCLUDE list of up to eight TCP/IP stack names to override the list of stacks supplied by the operating system (z/OS) or through SYSGEN (and z/VM).

All the stack-names that are specified on these SYSIDMS parameters are excluded from the runtime list of stacks. INCLUDE_TCP/IP_STACK and EXCLUDE_TCP/IP_STACK are mutually exclusive parameters and support wildcards.

FILE_BUF=*ddname*=*nnnnn*

Allows users to increase the number of pages in a buffer used by a specific file for a local mode job without having to change the DMCL. In CV, a DCMT command can be used to alter the number of pages in a buffer. The FILE_BUF parameter provides a similar capability for local mode jobs. If specified, the number of pages in the buffer pool associated with the specified file is increased by *nnnnn*.

This parameter can be used to tune PREFETCH processing by allowing the local mode user to increase the number of pages in a specific buffer for a job and thereby maximize the benefit of prefetch processing.

IDMSQSAM=ON|OFF

ON activates the IDMSQSAM facility (sequential access for look-ahead database reads).

Default: OFF

IGNORE_SYSTRK_DMCL=ON|OFF

Specifies whether to disable building the runtime DMCL from the SYSTRK file and instead force the use of the DMCL load module.

ON

Specifies to disable use of SYSTRK for building the runtime DMCL.

OFF

Specifies to build the runtime DMCL from SYSTRK if the CV was not previously shutdown normally.

Default: OFF

INCLUDE_TCP/IP_STACK=stack-name

(z/OS and z/VM only) Creates an INCLUDE list of up to eight TCP/IP stack names to override the list of stacks supplied by the operating system (z/OS) or through SYSGEN (z/OS and z/VM). For z/VM only, it can be used to include a new stack.

All the stack-names that are not specified on these SYSIDMS parameters are excluded from the runtime list of stacks. INCLUDE_TCP/IP STACK and EXCLUDE_TCP/IP_STACK are mutually exclusive parameters and support wildcards.

JOURNAL=ON|OFF

Specifies whether journaling will be performed in local mode. OFF specifies that local mode journaling will not be performed, even if there are tape journals defined in the DMCL.

Default: ON

JRNLDTS=*yyyy-mm-dd-hh.mm.ss.nnnnnn*

This parameter provides a way to bypass a date time stamp mismatch problem between the DMCL and the journal files. The yyy-mm-dd-hh.mm.ss.nnnnnn is the date time stamp on the journal file. This should only be used if you know that the reason for the mismatch will not cause a problem. Inappropriate use of this parameter may cause database corruption.

LANG=*xxxxxxxxxxxxxxxxxxx*

Sets an alternate environment for DBCS support. This parameter is useful for local mode batch jobs and is equivalent to issuing the DCUF SET LANG command for online users.

Limits: 19 characters

LENGTH_PAGE=*nnn*

Specifies the maximum number of lines to be printed on a page.

Limits: 10-32,767 lines

Default: 60

LIMIT_PREFETCH=*nnn*

When a batch job is performing an area sweep, prefetch takes 3/4 of the number of buffers and divides by the number of records per track to determine the maximum number of start I/Os it can issue. If the number is greater than 100, a value of 100 is used. At the start of a job this can swamp the system with I/O requests.

By coding LIMIT_PREFETCH=nnn prefetch will only issue nnn I/Os at any time. If this number is greater than the value calculated by CA IDMS, the lower value will be used.

LOADAREA=ON|OFF

Specifies whether the dictionary load (DDLDCLOD) area is to be accessed when loading a module. If OFF is specified, the dictionary load area will not be accessed. You should specify OFF only when all load modules are linked into an OPSYS load library.

Default: ON

LOCAL=ON|OFF

Specifies whether a batch job is to execute in local mode. If ON is specified, all requests are processed locally even if an IDMSOPTI is link-edited with the program, or a SYSCTL file is specified in the JCL.

Default: OFF

LOCAL_DYNAMIC_ALLOCATION= ON|OFF

OFF directs a local CA IDMS batch job to ignore any DSN information defined in the DMCL for database files, and requires that the DSN information be included in the JCL in order to access a database file.

Default: ON

LOCAL_NOJOURNAL_RETRIEVAL

Specifies that local batch jobs not journal RETRIEVAL ONLY transactions.

LOCALPUR=ON|OFF

ON forces the purging of the local mode buffer pool whenever a transaction terminates.

This parameter addresses a change in the way local mode buffers are handled (between 10.21 and later releases). In release 10.21 a local mode job that had multiple transactions (run units) would have separate buffer pools for each transaction (and each transaction would have no knowledge of the others). When a transaction terminated its buffer pool would be purged. Starting in release 12.0, a local job with multiple transactions will have just one buffer pool shared by all transactions. When a transaction terminates the buffer remains unchanged until the last transaction terminates at which time the shared buffer pool is purged. To make the system purge the common buffer pool when each transaction terminates (and therefore mimic what happened in release 10.21), use the parm LOCALPUR=ON.

Default: OFF

Note: This parameter should only be specified if a compatibility problem is encountered, since there can be performance implications in specifying LOCALPUR=ON.

MSGDICT=ON|OFF

Specifies whether the dictionary message (SYSMSG) area is to be accessed in order to retrieve the text of messages. If OFF is specified, the dictionary message area is not accessed. OFF should be specified only when using a DMCL that does not contain the SYSMSG segment, such as during installation.

Default: ON

MULTIDSN=ON|OFF

(z/VSE only) ON specifies that tape files may span multiple volumes. At the end of a tape reel, EOF (end of file) or EOV (end of volume) prompts the user to specify an END OF JOB or an END OF VOLUME condition. OFF specifies that END OF JOB is automatically the condition at the end of a tape reel.

Default: OFF

NODENAME=*nodename*

For non-SQL applications running under the central version, identifies the DC/UCF system to bind to at run time.

PARM=*'parameter-string'*

Allows you to specify parameters typically specified in a JCL EXEC PARM statement. The format is the same as the IBM PARM parameter on the EXEC JCL statement. parameter-string can contain any 1 through 256 character parameter and can be specified on multiple lines.

PDAT_LOCATION=ANY|BELOW

Specifies whether the UCF line driver allocates PDAT storage above or below the 16 MB line. ANY, the default, allocates PDAT storage above the line. BELOW allocates PDAT storage below the line. BELOW must be specified if a back-end PDAT application runs with AMODE 24.

Default: ANY

DATE_SIMULATOR_SVC=nnn

Causes the CA IDMS CV or batch job to issue an SVC to call the DATE SIMULATOR and replaces the STCK (STORE CLOCK) commands as documented in the DATE SIMULATOR instructions provided by the Date Simulator vendor.

The purpose of this parameter is to allow clients to run their DATE SIMULATOR products without zapping the CA IDMS modules.

PREFETCH=ON|OFF

OFF overrides the default ON and prevents CA IDMS from prefetching database pages, the normal processing when an area or index sweep is detected. Specify OFF for a local batch job to prevent prefetching database pages for the job step. Specify OFF in the SYSIDMS file associated with a central version to prevent prefetching pages for all transactions running with the central version.

Default: ON

PREFETCH_BUF=*nnnnn*

Specifies the minimum number of pages in a buffer pool that must be present before CA IDMS will use prefetch processing for non-area sweep requests. This parameter applies to both local and central version environments.

PRMPT_MSG='*prompt_message*'

Customizes the WTOR (Write-To-Operator-With-Reply) message. Customizing this message lets you distinguish the different CVs that are running.

Enclose *prompt_message* in single (') or double (") quotes. You can write the message in free-form, but do not enter internal quotes.

Limits: The message can be 1 to 58 characters long (without quotes).

Default: If the parameter is not specified, the following message is displayed:

```
REPLY WITH REQUEST TO IDMS Vnnn
```

nnn is the system number, which is present regardless if the default or a customized message is displayed.

PROCTRACE=ON|OFF

ON activates a trace of key user blocks that participate in an SQL PROCEDURE call.

Default: OFF

QSAMAREA=*qsam-area-name*

Specifies the physical area in the DMCL for which the IDMSQSAM facility will do look-ahead reads. If this parameter is omitted and the IDMSQSAM=ON parameter is specified, the look-ahead reads will be performed on the first area that is accessed by the transaction.

Note: This parameter may also be used against journal files when executing the ARCHIVE JOURNAL utility by specifying QSAMAREA=ARCHIVE.JOURNAL in the utility's SYSIDMS file.

QSAMBUF#=*nnn*

(z/OS only) Specifies the number of buffers to use when the IDMSQSAM facility is active. QSAMBUF# enables you to set the number of QSAM buffers to be used without having to code JCL for the file being processed by IDMSQSAM.

If QSAMBUF# is not specified, the number of buffers is determined by the DCB parameter BUFNO=nnn, or defaults to 5 buffers.

Limits: 1-255

Default: 5 buffers

QSAMTRACE=ON|OFF

ON activates a trace of all the IDMSQSAM look-ahead I/O reads. This trace shows the name of the file(s) being accessed by IDMSQSAM, each RBN that is read using QSAM or BDAM (DAM/EXCP), and a summary of the number of RBN's read through QSAM and BDAM. It also shows the area being accessed and the number of OPSYS QSAM buffers being used as determined by the JCL.

Default: OFF

REREAD_SYSCTL=ON|OFF

ON directs local mode operations to reread the SYSCTL file for each new transaction. This allows you to do the following:

1. Include a SYSCTL in a batch job step's JCL.
2. Start a transaction that will execute under central version, based on the contents of the SYSCTL file.
3. Deallocate the SYSCTL file defined in the JCL.
4. Start another transaction to execute in local mode.

Default: OFF

ROLLBACK3490

Enables the ROLLBACK utility to process archive files residing on devices that do not support backward read, such as disk and 3490E devices.

SAVE_SQL_SYNTAX=ON|OFF

If ON is specified during CV startup, all dynamic SQL syntax is saved before it gets optimized. To do so, the SQL server module IDMSQSRV allocates a piece of long term storage each time an sLTE (that is a secondary LTE representing the SQL session) is acquired. That storage is anchored off the sLTE and is used to save the dynamic SQL syntax before it is processed by the syntax parser and optimized.

The storage is obtained only once per session and is re-used if there is more SQL syntax to save for the same session. The storage is released during the end of session processing just prior to freeing the sLTE.

You can obtain the saved SQL syntax by Exit 39 - SQL Syntax Collecting Exit.

Default: OFF

SCRATCH_IN_STORAGE=ON|OFF|ANY|XA|64-bit

Enables storage allocation from the operating system for scratch processing.

ON

Specifies the same as SCRATCH_IN_STORAGE=ANY.

OFF

Specifies to allocate the scratch area as defined in the DMCL.

ANY

Acquires 64-bit storage if possible. If the request to allocate 64-bit storage fails, XA storage is acquired.

XA

Acquires 31-bit storage.

64-bit

Acquires 64-bit storage. If the request to allocate 64-bit storage fails, no attempt to acquire XA storage is done.

Default: OFF

Note: Usage of 64-bit storage is controlled by the MEMLIMIT parameter of the JOB or EXEC JCL statement on z/OS, and by the MEMLIMIT option on the SYSDEF MEMOBJ command on z/VSE.

SCRATCH_LIMIT=*limit-with-unit*

Specifies the maximum amount of scratch storage. The system continues to allocate more storage for scratch processing until the sum of all allocations reaches the value specified by *limit-with-unit*. Enter a number in the range 1-32767 followed by a unit of KB (Kilobyte: 2**10), MB (Megabyte: 2**20), GB (Gigabyte: 2**30), TB (Terabyte: 2**40), or PB (Petabyte: 2**50).

The default value is determined as follows:

- If the DMCL contains a scratch area definition, the default is the number of pages in the area multiplied by the page size.
- If no scratch area is defined in the DMCL, the default is the size of the primary allocation plus 99 times the size of the secondary allocation.

SCRATCH_PRIMARY_EXTENT=*prim-size-with-unit*

Specifies the primary scratch allocation size. Enter a number in the range 1-32767 followed by a unit of KB (Kilobyte: 2**10), MB (Megabyte: 2**20), GB (Gigabyte: 2**30), TB (Terabyte: 2**40), or PB (Petabyte: 2**50).

The default value is determined as follows:

- If the DMCL contains a scratch area definition, the default value is the number of pages in the area multiplied by the page size.
- If no scratch area is defined in the DMCL, the system default value is 1 MB.

SCRATCH_SECONDARY_EXTENT=*sec-size-with-unit*

Specifies the amount of storage to allocate when all existing storage is in use. Enter a number in the range 1-32767 followed by a unit of KB (Kilobyte: 2**10), MB (Megabyte: 2**20), GB (Gigabyte: 2**30), TB (Terabyte: 2**40), or PB (Petabyte: 2**50).

The default size of the secondary allocation is equal to the size of the primary allocation.

SHUTDOWN_STALL_TIME=*nnn*

Causes the stall interval for active user tasks to be set to *nnn* seconds after a SHUTDOWN command is issued. If the parameter is not coded or an interval of 0 (zero) is specified, then there is no effect on SHUTDOWN processing. This parameter allows normal SHUTDONW processing even if a user task is in a long wait for an unavailable resource. The specified interval is used to force user tasks to be aborted if a given wait exceeds the interval even if the task normally has a longer STALL interval. This parameter affects online application tasks and ERUS tasks in either an INTERNAL or EXTERNAL wait. It does not have any effect on system tasks such as service drivers, line drivers, factotums, and helots.

SORT FIELD MAX LEN=n

n specifies the maximum sort key field length allowed by a sort.

Default (z/OS): 2000

Default (z/VSE): 256

SORTSIZE=ON|OFF

Directs whether CA IDMS utilities generate the SIZE= sort parameter card. Some sort packages cannot handle the SIZE= parameter. The default is OFF, which means that the SIZE= sort parameter is not generated.

Default: OFF

SQL_CACHE_ENTRIES=n

n specifies the maximum number of entries that will be used in the dynamic SQL cache. One entry holds one cached SQL statement. Specify 0 to disable caching. The maximum theoretical value is 2,147,483,647. The actual maximum depends on available memory.

Default: 200

SQL_INTLSORT=ON|OFF

Allows you to force the internal CA IDMS sort to be used in local mode. If ON is specified, an internal SORT rather than an operating system SORT will be performed on SQL commands issued in a local batch job that contain an ORDER BY clause. In many cases, an internal SORT is faster than an operating system SORT when you are not dealing with a large amount of data. OFF is the default, indicating an operating system SORT will be used.

Default: OFF

SQLTRACE=ON|OFF

ON activates a trace facility of all the SQL database requests made by an application.

Default: OFF

SYS_MSG=UPLOW|UPPER

UPPER directs CA IDMS to translate the text of internal #WTL messages to uppercase before being displayed at the output destination. The default is UPLOW. This allows the text of an internal #WTL message issued by CA software to be displayed in mixed case letters.

Specify UPPER under the following conditions:

- In local batch jobs to translate any internal #WTL messages issued by CA software to uppercase for that job step.
- In the SYSIDMS file associated with a central version to translate any internal #WTL messages issued by CA software to uppercase for that CV region.

SYSCTL=*ddname*

Specifies an alternate ddname for the SYSCTL file (other than the default ddname of SYSCTL).

SYSTRACE=ON|OFF

Controls whether basic system tracing is enabled.

ON

Enables basic system tracing.

Note: If basic system tracing is enabled by the SYSIDMS SYSTRACE parameter, it remains enabled for a system even if its system definition indicates that SYSTRACE is OFF.

OFF

Does not enable basic system tracing.

SYSTRK_DDNAME_PREFIX=*xxxxxxx*

Specifies the DDname prefix to be used for referencing SYSTRK files in execution JCL.

xxxxxxx

Specifies the 1- to 7-character DDname prefix.

Default: SYSTRK

TCP/IP_STACK_1=tcp/ip-stack-name

TCP/IP_STACK_2=tcp/ip-stack-name

TCP/IP_STACK_3=tcp/ip-stack-name

TCP/IP_STACK_4=tcp/ip-stack-name

TCP/IP_STACK_5=tcp/ip-stack-name

TCP/IP_STACK_6=tcp/ip-stack-name

TCP/IP_STACK_7=tcp/ip-stack-name

TCP/IP_STACK_8=tcp/ip-stack-name

In r16, the default TCP/IP stack name was extracted from the SYSTCPD file, and these 8 SYSIDMS parameters could be used to include 8 additional stacks to the CA IDMS TCP/IP environment. In r17, these parameters are still supported for compatibility reasons, but the method to define the TCP/IP stacks using the INCLUDE STACK clause in SYSGEN, or the two new SYSIDMS parameters INCLUDE_TCP/IP_STACK and EXCLUDE_TCP/IP_STACK will take precedence on these r16 definitions.

TCP/IP_STATUS=ON|OFF

Specifies the default status of TCP/IP support at startup. It overwrites the DEFAULT STATUS value defined to SYSGEN. OFF disables TCP/IP support for CA IDMS at startup. ON enables TCP/IP support.

TRACE_TABLE_SIZE=table-size KB | MB

Specifies the size of the system trace table in kilobytes (KB) or megabytes (MB).

Limit: 0–9999

Default: 0

Note: If basic system tracing is enabled by the SYSIDMS SYSTRACE parameter and the table size is 0, the table size is changed to 4 MB.

Any non-zero table size established by a SYSIDMS parameter overrides the trace table size specified in the system definition.

TRANSACTION_SHARING=ON|OFF

ON activates the 'Transaction Sharing' feature for all database activity used in a batch application.

Default: OFF

UPPER=INPUT|OUPUT|BOTH|OFF

Specifies whether input and/or output files will be converted to uppercase:

INPUT

Converts SYSIPT input files to uppercase.

OUTPUT

Converts SYSLST output files to uppercase.

BOTH

Converts both SYSIPT input files and SYSLST output files to uppercase.

OFF

Does not convert SYSIPT input files or SYSLST output files to uppercase.

Default: OFF

USERCAT=ON|OFF

Specifies whether the user catalog is to be accessed. Specify OFF only when formatting the user catalog or when the DMCL does not have access to a user catalog.

Default: ON

WIDTH_PAGE=*nnn*

Specifies a maximum number of characters to be printed on a SYSLST output line.

Limits: 71-132

Default: 132

WORK=n

CA IDMS utilities that generate SORT input cards for use within a VSE environment typically hardcode the number of SORTWK files to be allocated. This value can be insufficient for the amount of data to be sorted. This parameter allows sites to specify the number of SORTWK files to be allocated by their SORT.

Limits: 1-9

Note: Any other values are ignored and the default value of the utility is used instead.

XA_SCRATCH=ON|OFF

Specifies whether scratch space will be allocated out of XA storage or not. OFF, the default, indicates that a scratch file will be used.

Default: OFF

Note: XA_SCRATCH=ON/OFF is maintained for compatibility reasons only. Use SCRATCH_IN_STORAGE instead.

BUFFERSTAT Report Field Descriptions

The following table gives the descriptions for the fields displayed on the report produced by the BUFFERSTAT SYSIDMS parameter.

***** Buffer Name *****

The name of the buffer from the DMCL which has been opened during the processing of this job. Only those buffers which are open or have had some I/O activity against them will appear in this report.

***** Pages *****

The number of pages allocated to the buffer. This is the actual number of pages 'in use' by this buffer for this job. This number is the total from the a) DMCL Local Mode Buffer Pages nnn, or b) JCL DCB=BUFNO=nnn, or c) SYSIDMS Parm FILE_BUF=ddname=nnn.

***** Prefetch Minimum *****

The minimum number of buffers 'in use' needed to allow Prefetch to operate from Random access verbs (that is, non-area sweep processing).

DB Page Requests

The number of times IDMSDBMS requests a database page from the buffer pool by calling IDMSDBIO.

Sequential Area Request

Of the DB Page requests in the 'DB Page Requests' count, how many were GET/NEXT/PRIOR in AREA verbs.

Random Request

Of the DB Page requests in the 'DB Page Requests' count, how many were not GET/NEXT/PRIOR in AREA verbs.

Found in Buffer

Of the DB Page requests in the 'DB Page Requests' count, how many DB pages were already in the buffer pool and did not require an I/O.

Not Found in Buffer

Of the DB Page requests in the 'DB Page Requests' count, how many DB pages were not in the buffer pool and therefore required an I/O.

Buffer 'hit' Ratio

Calculated as 'Found in Buffer' times 100, divided by the DB Page Requests value.

Found in Pref Buffer

Found in Prefetch Buffer. Of the 'DB Page Requests' count, how many DB pages were found in the buffer pool that had been read by a Prefetch Read?

Found in Cache

Of the 'DB Page Requests' count, how many DB pages were found in the shared cache?

Total DB Pages Read

The total number of DB pages read by both Prefetch and standard I/O. This is not the number of I/O's or EXCP's, but the number of DB pages read into a buffer as a standalone I/O, 'Start I/O - Reads', or as part of a Prefetch I/O 'Pages Read - Prefetch'.

DB Pg Req:Tot Pages Read

The number of 'DB Page Requests' count divided by the 'Total DB Pages Read' count.

DB Pg Req:Strt I/O Read

The number of 'DB Page Requests' count divided by the 'Start I/O Reads' count.

NonPrefetch I/O Rqst

The number of standard of non-Prefetch I/O requests. This is the number of the 'DB Pages Request' count which are 'Not found in Buffer' and were not eligible for Prefetch. Prefetch was either not allowed, turned off, or the request was started by a 'Random Request' for which the '*** Prefetch Minimum ***' is higher than the number of '*** Pages ***'.

Start I/O - Reads

The number of standard or non-Prefetch I/O reads. This number is the result of 'DB Page Requests' which were 'Not found in Buffer' and Prefetch is turned off, or the request was started by a 'Random Request' for which the '*** Prefetch Minimum ***' is higher than the number of '*** Pages ***' count, or the request is not eligible for Prefetch processing. Each of these will show up as 1 EXCP.

Start I/O - Writes

The number of writes started against the database. Each of these will show up as 1 EXCP.

Read forces Write

In order to read a DB page into the buffer, a DB page had to be written out to the database first (based on the least recently used algorithm). When this occurs, Prefetch is effectively turned off.

Prefetch Requests

With Prefetch operating, the number of DB page requests 'Not found in Buffer', which were eligible for Prefetch processing.

Sequential Area Request

Of the 'Prefetch Requests' count, how many were GET NEXT/PRIOR in AREA type verbs.

Random Request

Of the 'Prefetch Requests' count, how many were not GET NEXT/PRIOR in AREA type verbs.

Pref Req Denied:Buffers

Of the 'Prefetch Requests' count, how many did not use Prefetch due to too many buffer pages with the 'must write switch' on (over 1/2 the number of pages in the buffer pool).

Start I/O - Prefetch

The number of the 'Prefetch Requests' count that actually "Start an I/O" or "Start Subchannel". Each of these will show up as 1 EXCP.

Pages Read - Prefetch

The number of DB pages that were "carried" with every 'Start I/O - Prefetch'. This number plus the 'Start I/O - Reads' will equal the 'Total DB Pages Read' count.

Pref Strt I/O:Pref Req

The 'Start I/O - Prefetch' count divided by the 'Prefetch Requests' count.

Pref Pages:Pref Strt I/O

The 'Pages Read - Prefetch' count divided by the 'Prefetch Requests' count.

Pref Pages:Pref Strt I/O

The 'Pages Read - Prefetch' count divided by the 'Start I/O - Prefetch' count. This value shows how effective the Prefetch operation is. Compare this number to the "pages per track" to see how effective each Prefetch I/O is. If this number is around 3/1 or less, you probably will not see enough improvement in performance to warrant using Prefetch.

For each file in use, there will be a set of counts:

- Filename—The DDname of the file using this buffer. Only the files that are open will show up in this report.
- Pgs read—The number of DB pages read from this file. This is not the number of I/Os or EXCPs.
- Written—The number of DB pages written to this file. This is the number of I/Os or EXCPs.
- In buffer—The number of 'DB Page Requests' that were found in the buffer pool which this file maps to.
- In prefetch—The number of 'DB Page Requests' that were found in the buffer pool which this file maps to due to Prefetch.

z/VSE File Parameters

The following parameters can be used to override default values for z/VSE work files and for the system logical units SYSRDR, SYSIPT, SYSPCH, and SYSLST in batch jobs only.

FILENAME=*file-name*

Specifies the name of the file whose attributes are to be overridden by the following SYSIDMS parameters.

BLKSIZE=*block-size*

Specifies the block size for a file. BLKSIZE and BLOCKS are mutually exclusive parameters.

BLOCKS=*block-count*

Specifies a blocking factor for a file. BLKSIZE and BLOCKS are mutually exclusive parameters.

DEVADDR=SYSxxx

Specifies a device address for a tape file (SYSIPT, SYSLST, SYSRDR, SYSPCH, or SYSlogical-unit-number).

FILABL=STD/NO

Specifies a no-label option for a tape file. FILABL=STD is the default.

Default: FILABL=STD

FILETYPE=T/D/I

Specifies a file type of tape, disk, or file independent.

LRECL=*logical-record-size*

Specifies the logical record size for a file.

RECFM=F/V

Specifies if the current file contains fixed-length or variable-length records. V is the default.

Default: V

REWIND=YES/NO/UNLOAD

Specifies the position of a tape file when it is opened or closed. REWIND=UNLOAD is the default.

Default: REWIND=UNLOAD

Appendix A: Using the CommandFacility as a Subprogram

This section contains the following topics:

Command Facility Subprogram 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.

Storage type	Advantages/disadvantages
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 Work-Area File.

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.

Last Statement—END

If an END command is not passed to the compiler at each IDMSOCF invocation, a suspended session is leftover. Thus, causing allocated storage to remain attached to the LTE until the user signs off or the LTE is recycled. It is therefore recommended that the last command in the input file be END to prevent LTE storage build-up.

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).

Compiler Interface Parameter List

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

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.

Error code	Description
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.

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

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

Parameter	Information passed
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'

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.

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.           Month dd, yyyy.
 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 +4.
****************************************************************
*    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