CA IDMS - 19.0
Using Common Facilities

Date: 15-Jan-2018
This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by CA at any time. This Documentation is proprietary information of CA and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA.

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

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

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

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

The manufacturer of this Documentation is CA.

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

Copyright © 2017 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.
# Table of Contents

## Command Facility
- Types of Statements You Can Submit ......................................................... 8
- Coding Considerations - Command Facility .................................................. 9
- Special Coding for Command Facility Statements ......................................... 10
- Connecting to a Dictionary ............................................................................ 11
- Using SET OPTIONS to Select Options ....................................................... 12

## IDMSBCF Batch Processing
- IDMSBCF JCL for z/OS .................................................................................. 24
- IDMSBCF Commands for z/VM .................................................................... 26
- IDMSBCF JCL for z/VSE ............................................................................... 28

## Online Processing
- Beginning an OCF Session ............................................................................ 33
- Conducting an Online Session ..................................................................... 34
- Ending an OCF Session ................................................................................ 35
- Executing Statements Dynamically ............................................................... 36
- Executing Statements Stored in Modules (OCFX) ......................................... 36
  - OCFX Status Codes .................................................................................. 38
- Creating, Editing, and Saving Modules .......................................................... 39

## Using the Online Text Editor
- Online Compilers .......................................................................................... 40
- The IDD Menu Facility .................................................................................. 42

## Using Top-line Commands
- Summary of Top-line Commands ................................................................. 46
- APPLY ............................................................................................................ 47
- CLEAR .......................................................................................................... 47
- DELETE ALL ................................................................................................. 47
Using Line Commands .............................................................................. 61
- Copying Lines ...................................................................................... 61
- Deleting Lines ..................................................................................... 62
- Entering Block Commands .................................................................... 63
- Moving Lines ...................................................................................... 64
- Repeating Lines .................................................................................. 65
- Repositioning Lines on the Screen ........................................................ 66

Using Control Keys .................................................................................. 66
- Simulating PF Keys ................................................................................ 67
- Using OCF and Online Compiler Control Keys .......................................... 68
- Using IDD Menu Facility Global Control Keys .......................................... 68

Using Transfer Control Facility (TCF) ....................................................... 69
- Tools You Can Invoke from TCF ............................................................... 69
- Examples: Transfer Control Facility ......................................................... 70
  - Transferring from ADSC to the Online Map Compiler ............................ 71
  - Transferring from MAPC to IDD ............................................................ 72
  - Transferring from IDD back to MAPC .................................................. 72
  - Transferring to TCF from MAPC .......................................................... 73
Using Common Facilities

The Selection Screen Is Displayed ........................................................................................................... 74
Your Suspended ADSC Session Is Resumed ........................................................................................... 74

Basics of Using TCF ................................................................................................................................ 75
Using the TCF ........................................................................................................................................... 75
Transferring Between Development Tools .............................................................................................. 78
Invoking a Development Tool .................................................................................................................. 78
Exiting from a Development Tool ........................................................................................................... 79
Exiting from TCF ..................................................................................................................................... 80

TCF Selection Screen ............................................................................................................................... 81

Development Tools under TCF ................................................................................................................ 85
CA ADS and the Online Map Compiler .................................................................................................... 85
  Select the SWITCH Activity from the Main Menu .................................................................................. 85
  Transferring from CA ADS Compilers or the Online Map Compiler ................................................... 87
CA IDMS Online Compilers ...................................................................................................................... 87
CA OLQ under TCF .................................................................................................................................. 89
ASF .......................................................................................................................................................... 90
IDD ......................................................................................................................................................... 91

Using the Command Facility as a Subprogram ....................................................................................... 93
Command Facility Subprogram Basics .................................................................................................... 94
Compiler Interface Parameters ................................................................................................................. 94
Work-area File ......................................................................................................................................... 97
Sample Program that Calls the Command Facility ................................................................................... 97
Using Common Facilities

This section 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.

For more information, see the following topics:
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
For more information, see the following topics:
- Coding Considerations - Command Facility (see page 9)
- Special Coding for Command Facility Statements (see page 10)
- Connecting to a Dictionary (see page 11)
- Using SET OPTIONS to Select Options (see page 12)

## Coding Considerations - Command Facility

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

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input</strong></td>
<td>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 (<a href="https://docops.ca.com/display/IDMS19/SYSIDMS+Parameter+File">https://docops.ca.com/display/IDMS19/SYSIDMS+Parameter+File</a>) section. 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.</td>
</tr>
<tr>
<td><strong>Syntax</strong></td>
<td>Use the syntax rules that apply to the type of statement: Physical DDL -- See the CA IDMS Database Administration Section. Logical DDL (SQL-defined databases) -- See the CA IDMS SQL Reference Section. SQL DML -- See the CA IDMS SQL Reference Section. Utilities -- See the CA IDMS Utilities Section. Security administration -- See the CA IDMS Security Administration Section.</td>
</tr>
<tr>
<td><strong>Statement terminator</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Comments</strong></td>
<td>Begin comments with a double hyphen (--) Comments continue to the end of an input line.</td>
</tr>
<tr>
<td><strong>Connecting to a dictionary</strong></td>
<td>Use a CONNECT statement in a batch or online command facility session (see Connecting to a Dictionary (see page 11))</td>
</tr>
<tr>
<td><strong>Session control</strong></td>
<td>Use the SET OPTIONS statement (see Using SET OPTIONS to Select Options (see page 12))</td>
</tr>
</tbody>
</table>
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.

```plaintext
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.

  ```plaintext
  Note: For more information on SYSIDMS parameters, see SYSIDMS Parameter File (https://docops.ca.com/display/IDMS19/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 on DCUFs, see the *CA IDMS System Tasks and Operator Commands Section*. For more information on the user signon profile, see the *CA IDMS Security Administration Section*. For more information on COMMAND FACILITY SIGNON, see the *CA IDMS IDD DDDL Reference Section*. 
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
STATUS OFF
       ON
COMPRESS ON
       OFF
COLUMN WRAP OFF
       ON

Expansion of io-option
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 on 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.

<table>
<thead>
<tr>
<th>Option</th>
<th>IDMBSFC</th>
<th>OCF</th>
<th>OCFX</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTO COM MIT ON (default)</td>
<td>Executes a COMMIT WORK RELEASE after the last statement in the input stream.</td>
<td>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.</td>
<td>Executes a COMMIT WORK RELEASE at the end of the module.</td>
</tr>
<tr>
<td>AUTO COM MIT ON CONTINUE</td>
<td>N/A</td>
<td>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.</td>
<td>N/A</td>
</tr>
<tr>
<td>AUTO COMMIT OFF</td>
<td>No automatic COMMIT occurs; you must specify COMMIT WORK RELEASE sectionly.</td>
<td>-- A release is issued at the end of processing, and this may cause a rollback if any work has not been committed.</td>
<td></td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Option</th>
<th>IDMSBCF</th>
<th>OCF</th>
<th>OCFX</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON ERROR</td>
<td>Continues executing.</td>
<td>Continues executing.</td>
<td>Continues executing.</td>
</tr>
<tr>
<td>CONTINUE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(default)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ON ERROR</td>
<td>Ends job step and session.</td>
<td>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.</td>
<td>Ends execution of commands in the module and ends the session. Proceeds with the AUTOCOMMIT option.</td>
</tr>
<tr>
<td>END</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ON ERROR</td>
<td>Issues a ROLLBACK RELEASE and ends job step.</td>
<td>Issues a ROLLBACK RELEASE and ends the SQL session. Does not end the OCF session.</td>
<td>Issues a ROLLBACK RELEASE and ends the execution of the module.</td>
</tr>
<tr>
<td>ROLLBACK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACK</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: ROLLBACK does not apply to utility statements.

- **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;
```

<table>
<thead>
<tr>
<th>EMP_LNAME</th>
<th>EMP_FNAME</th>
<th>DEPT_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brooks</td>
<td>John</td>
<td>3510</td>
</tr>
<tr>
<td>Park</td>
<td>Deborah</td>
<td>2210</td>
</tr>
<tr>
<td>Smith</td>
<td>Carl</td>
<td>3530</td>
</tr>
</tbody>
</table>
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.

```
SET OPTIONS WIDTH CHAR 10;
```

```
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 Mechanic        20500.00  45500.00  yyyy-mm-dd
    5555 Salesperson        15000.00  39500.00  yyyy-mm-dd
    4123 Recruiter          17500.00  28000.00  yyyy-mm-dd
    4025 Writer - Mktng    15500.00  25000.00  yyyy-mm-dd
    4023 Accountant         22000.00  60000.00  yyyy-mm-dd
    8001 Vice President     45000.00  60000.00  yyyy-mm-dd
    2077 Purch Clerk        8500.00   15000.00  yyyy-mm-dd
    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 script 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(' || '||''||' || trim(VALUE('
        || SUBSTR('CAST(' || trim(TYPE) || ' as char(10))'' || trim(NAME) || ' ' ||
        || SUBSTR('CAST(' || trim(TYPE) || ' as char(10))'' || trim(TREND) || ' as char(10))'' || trim(NAME) || ' ' ||
        || SUBSTR('CAST(' || trim(TYPE) || ' as char(10))'' || trim(NAME) || ' ' ||
        || SUBSTR('CAST(' || trim(TYPE) || ' as char(10))'' || trim(NAME) || ' ' ||
        || SUBSTR('CAST(' || trim(TYPE) || ' as char(10))'' || trim(NAME) || ' ' ||
FROM SYSTEM.COLUMN, USERA01.CopyTabParm
WHERE TABLE  = SrcTable
      AND SCHEMA = SrcSchema
UNION
SELECT '' );'' FROM '
      || trim(SrcSchema) || '.*' || trim(SrcTable) || ';''
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( ' || TRIM(VALUE(CAST(EMP_ID as char(10)),'NULL')) ||','||TRIM(VALUE(CAST(MANAGER_ID as char(10)),'NULL')) ||','||TRIM(VALUE(''''EMP_FNAME ||'''','NULL')) ||','||TRIM(VALUE(''''EMP_LNAME ||'''','NULL')) ||','||TRIM(VALUE(CAST(DEPT_ID as char(10)),'NULL')) ||','||TRIM(VALUE(''''STREET ||'''','NULL')) ||','||TRIM(VALUE(''''CITY ||'''','NULL')) ||','||TRIM(VALUE(''''STATE ||'''','NULL')) ||','||TRIM(VALUE(''''ZIP_CODE ||'''','NULL')) ||','||TRIM(VALUE(''''PHONE ||'''','NULL')) ||','||TRIM(VALUE(''''STATUS ||'''','NULL')) ||','||TRIM(VALUE(CAST(SS_NUMBER as char(10)),'NULL')) ||','||TRIM(VALUE(''''START_DATE as char(10))||''','NULL')) ||','||TRIM(VALUE(''''TERMINATION_DATE as char(10))||''','NULL')) ||','||TRIM(VALUE(''''BIRTH_DATE as char(10))||''','NULL')) ||');' from DEMOEMPL.EMPLOYEE;

Load OUTPUT

insert into USERA01.EMPLOYEE VALUES( 2299,NULL,'Samuel Spade','4600','47 London St','Canton','MA','02020',NULL,'L',33892200,'1991-02-04','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','1967-10-28');
insert into USERA01.EMPLOYEE VALUES( 4773,3082,'Janice Dexter','399 Pine St','Medford','MA','02432',5083847566,'A',89675632,'1997-06-14','1969-11-19');
... ...

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
IDMSBCF Batch Processing

The Batch Command Facility (IDMSBCF) is the batch component of the command facility. To submit statements in a batch environment, you enter the statements in an IDMSBCF job stream.

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 (see page 12).

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.
CA IDMS - 19.0

- 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.

For more information, see the following topics:
- IDMSBCF JCL for z/OS (see page 24)
- IDMSBCF Commands for z/VM (see page 26)
- IDMSBCF JCL for z/VSE (see page 28)

IDMSBCF 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.cagload,DISP=SHR
//sysctl DD DSN=idms.sysctl,DISP=SHR
//dcmg DD DSN=idms.sysmsg.ddlcmg,DISP=SHR
//SYSLST DD SYSOUT=A
//SYSIDMS DD *
DMCL=dml-name DICTNAME=dictionary-name Other SYSIDMS parameters, as appropriate/
//SYSIPT DD *Statements (including optional SET OPTIONS statements)/+
```

⚠️ **Note:** For more information on SYSIDMS parameters, see SYSIDMS Parameter File ([link](https://docops.ca.com/display/IDMS19/SYSIDMS+Parameter+File)).

<table>
<thead>
<tr>
<th>Stepname</th>
<th>Name of the job step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>idms.dba.</td>
<td>Data set name of the load library containing the DMCL and database name table load modules</td>
<td></td>
</tr>
<tr>
<td>loadlib</td>
<td></td>
<td></td>
</tr>
<tr>
<td>idms.</td>
<td>Data set name of the load library containing customized CA IDMS executable load modules</td>
<td></td>
</tr>
<tr>
<td>custom.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>loadlib</td>
<td></td>
<td></td>
</tr>
<tr>
<td>idms.</td>
<td>Data set name of the load library containing the CA IDMS executable modules that do not require customization</td>
<td></td>
</tr>
<tr>
<td>cagload</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sysctl</td>
<td>Ddname of the SYSCTL file</td>
<td></td>
</tr>
<tr>
<td>idms.sysctl</td>
<td>Data set name of the SYSCTL file (the link to the central version)</td>
<td></td>
</tr>
<tr>
<td>dcmg</td>
<td>Ddname of the system message (DDLDCMSG) area</td>
<td></td>
</tr>
<tr>
<td>idms.</td>
<td>Data set name of the system message (DDLDCMSG) area</td>
<td></td>
</tr>
<tr>
<td>sysmsg.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dddlcmg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dml-name</td>
<td>Name of the DMCL</td>
<td></td>
</tr>
</tbody>
</table>
dictionary-name: Name of the dictionary to access
You can specify DICTNAME as a SYSIDMS parameter or you can use the CONNECT TO dict name 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:

```plaintext
//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:

```plaintext
//dcat DD DSN=idms.system.ddlcata,DISP=SHR
//dcatl DD DSN=idms.system.ddlcata,DISP=SHR
//dcatx DD DSN=idms.system.ddlcata,DISP=SHR
```

dcat Ddbname of the system dictionary catalog (DDLCAT) area
idms.system.ddlcata Data set name of the system dictionary catalog (DDLCAT) area
dcatl Ddbname of the system dictionary catalog load (DDLCATLOD) area
idms.system.ddlcatalod Data set name of the system dictionary catalog load (DDLCATLOD) area
dcatx Ddbname of the system dictionary catalog index (DDLCATX) area
idms.system.ddlcatax 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:

```plaintext
//sqlld DD DSN=idms.syssql.ddlcat,DISP=SHR
//sqllod DD DSN=idms.syssql.ddlcatlod,DISP=SHR
//sqlldx DD DSN=idms.syssql.ddlcatx,DISP=SHR
//userdb DD DSN=user.userdb,DISP=SHR
```

sqlld Ddbname of the SQL catalog area
idms.syssql.ddlcat Data set name of the SQL catalog area
sqllod Ddbname of the SQL catalog load area
idms.syssql.ddlcatlod Data set name of the SQL catalog load area
sqlldx Ddbname of the SQL catalog index area
idms.syssql.ddlcatx Data set name of the SQL catalog index area
userdb Ddbname of the user database file
user.userdb Data set name of the user database file

If You Submit Utility Statements

For more information on the file assignments specific to utility statements, see the CA IDMS Utilities Section.
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.dldlocscr,DISP=SHR
//secdd DD DSN=idms.sysuser.ddlsec,DISP=SHR
//dictdb DD DSN=idms.appdict.ddldml,DISP=SHR
//sysjrnl DD DSN=idms.tapejrnl,DISP=SHR
```

dclscr
Ddname of the local mode scratch (DDLOCSCR) area

idms.sysloc.dldlocscr
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 (DDLDM) area

idms.appdict.ddldml
Data set name of the application dictionary definition (DDLDM) area

dlodd
Ddname of the application dictionary definition load (DLDCLOD) area

idms.appdict.ddldclod
Data set name of the application dictionary definition load (DLDCLOD) 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**

IDMSBCF Commands for z/VM

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

**IDMSBCF (z/VM)**

```latex
FILEDEF SYSIPT DISK sysipt data a (RECFM F LRECL ppp BLKSIZE pppEXEC IDMSFD ORSUN IDMSBCF)
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysipt data a</td>
<td>Filename, type, and mode of the file containing source statements</td>
</tr>
<tr>
<td>ppp</td>
<td>Record length of the input data file</td>
</tr>
<tr>
<td>nnn</td>
<td>Block size of the input data file</td>
</tr>
</tbody>
</table>

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 on SYSIDMS parameters, see [SYSIDMS Parameter File](https://docops.ca.com/display/IDMS19/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 .
```
IDMSBCF 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,lllll// ASSGN SYS020,DISK,VOL=nnnnnn,SHR
// EXEC IDMSBCFSOURCE statements/*

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

**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 on SYSIDMS parameters, see [SYSIDMS Parameter File](https://docops.ca.com/display/IDMS19/SYSIDMS+Parameter+File).

**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

<table>
<thead>
<tr>
<th>sysjrnl</th>
<th>Name of the tape journal file</th>
</tr>
</thead>
<tbody>
<tr>
<td>idms.tapejrnl</td>
<td>ID of the tape journal file</td>
</tr>
<tr>
<td>f</td>
<td>File number of the tape journal file</td>
</tr>
<tr>
<td>SYS009</td>
<td>Logical unit assignment for the journal file</td>
</tr>
</tbody>
</table>

**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.

```plaintext
* -------- 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 dccml,'idms.system.dccml',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,101
// ASSGN SYSnnn,DISK, VOL=nnnnnn, SHR
// DLBL dccatlod,'idms.system.dccatlod',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,21
// 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 dcm,'idms.system.dcm',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,135
// ASSGN SYSnnn,DISK, VOL=nnnnnn, SHR
// DLBL dcmmsg,'idms.sysmsg.dcmmsg',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,201
// ASSGN SYSnnn,DISK, VOL=nnnnnn, SHR
// DLBL dcmmsg,'idms.sysmsg.dcmmsg',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,6
// ASSGN SYSnnn,DISK, VOL=nnnnnn, SHR
// DLBL dclo,'idms.system.dclo',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,401
// ASSGN SYSnnn,DISK, VOL=nnnnnn, SHR
// DLBL dcloc,'idms.system.dcloc',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,401
// ASSGN SYSnnn,DISK, VOL=nnnnnn, SHR
// DLBL dcrun,'idms.system.dcrun',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,68
// ASSGN SYSnnn,DISK, VOL=nnnnnn, SHR
// DLBL dcrun,'idms.system.dcrun',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,135
// ASSGN SYSnnn,DISK, VOL=nnnnnn, SHR
// DLBL dcs,'idms.system.dcs',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,135
// ASSGN SYSnnn,DISK, VOL=nnnnnn, SHR
// DLBL dcs,'idms.system.dcs',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,201
// ASSGN SYSnnn,DISK, VOL=nnnnnn, SHR
// DLBL dcs,'idms.system.dcs',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 dirlld,'idms.sysdirl.ddldlod',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 inxdem,'idms.sqldemo.inxdemo',2099/365,DA
// EXTENT SYSnnn,nnnnnn,,ssss,11
// ASSGN SYSnnn,DISK, VOL=nnnnnn, SHR
// DLBL infodem,'idms.sqldemo.infodem',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 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
```
### CA IDMS - 19.0

```plaintext
// EXTENT SYSnnn,nnnnn,,ssss,2
// ASSGN SYSnnn,DISK,VOL=nnnnn,SHR
// DLBL secdd,'idms.sysuser.ddsec',2099/365,DA
// EXTENT SYSnnn,nnnnn,,ssss,26
// ASSGN SYSnnn,DISK,VOL=nnnnn,SHR
// DLBL dictdb,'idms.appldict.ddldml',2099/365,DA
// EXTENT SYSnnn,nnnnn,,ssss,51
// ASSGN SYSnnn,DISK,VOL=nnnnn,SHR
// DLBL dloddb,'idms.appldict.ddldml',2099/365,DA
// EXTENT SYSnnn,nnnnn,,ssss,26
// ASSGN SYSnnn,DISK,VOL=nnnnn,SHR
// DLBL sqllod,'idms.syssql.ddldml',2099/365,DA
// EXTENT SYSnnn,nnnnn,,ssss,101
// ASSGN SYSnnn,DISK,VOL=nnnnn,SHR
// DLBL sqllod,'idms.syssql.ddldml',2099/365,DA
// EXTENT SYSnnn,nnnnn,,ssss,51
// ASSGN SYSnnn,DISK,VOL=nnnnn,SHR
// DLBL sqllod,'idms.syssql.ddldml',2099/365,DA
// EXTENT SYSnnn,nnnnn,,ssss,26
// ASSGN SYSnnn,DISK,VOL=nnnnn,SHR
// DLBL asfdml,'idms.asfdict.ddldml',2099/365,DA
// EXTENT SYSnnn,nnnnn,,ssss,201
// ASSGN SYSnnn,DISK,VOL=nnnnn,SHR
// DLBL asflod,'idms.asfdict.ddldml',2099/365,DA
// EXTENT SYSnnn,nnnnn,,ssss,401
// ASSGN SYSnnn,DISK,VOL=nnnnn,SHR
// DLBL asfdml,'idms.asfdict.ddldml',2099/365,DA
// EXTENT SYSnnn,nnnnn,,ssss,201
// ASSGN SYSnnn,DISK,VOL=nnnnn,SHR
// DLBL ASFDEFN,'idms.asfdict.asmdefn',2099/365,DA
// EXTENT SYSnnn,nnnnn,,ssss,101
// ASSGN SYSnnn,DISK,VOL=nnnnn,SHR
// DLBL j1jrnl,'idms.j1jrnl',2099/365,DA
// EXTENT SYSnnn,nnnnn,,ssss,54
// ASSGN SYSnnn,DISK,VOL=nnnnn,SHR
// DLBL j2jrnl,'idms.j2jrnl',2099/365,DA
// EXTENT SYSnnn,nnnnn,,ssss,54
// ASSGN SYSnnn,DISK,VOL=nnnnn,SHR
// DLBL j3jrnl,'idms.j3jrnl',2099/365,DA
// EXTENT SYSnnn,nnnnn,,ssss,54
// ASSGN SYSnnn,DISK,VOL=nnnnn,SHR
// DLBL SYSIDMS,'#SYSIPT',0,SD
/+
/*

<table>
<thead>
<tr>
<th><strong>idmslib.sublib</strong></th>
<th>Name of the sublibrary within the library containing CA IDMS modules</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>user.sublib</strong></td>
<td>Name of the sublibrary within the library containing user modules</td>
</tr>
<tr>
<td><strong>idmslib</strong></td>
<td>Name of the file containing CA IDMS modules</td>
</tr>
<tr>
<td><strong>idms.library</strong></td>
<td>ID associated with the file containing CA IDMS modules</td>
</tr>
<tr>
<td><strong>SYSnnn</strong></td>
<td>Logical unit of the volume for which the extent is effective</td>
</tr>
<tr>
<td><strong>nnnnnn</strong></td>
<td>Volume serial identifier of appropriate disk volume</td>
</tr>
<tr>
<td><strong>ssss</strong></td>
<td>Starting track (CKD) or block (FBA) of disk extent</td>
</tr>
<tr>
<td><strong>dccat</strong></td>
<td>Filename of the system dictionary catalog (DDLCAT) area</td>
</tr>
<tr>
<td><strong>idms.system.dccat</strong></td>
<td>ID of the system dictionary catalog (DDLCAT) area</td>
</tr>
<tr>
<td><strong>dccatl</strong></td>
<td>Filename of the system dictionary catalog load (DDLCATLOD) area</td>
</tr>
<tr>
<td><strong>idms.system.dccatlod</strong></td>
<td>ID of the system dictionary catalog load (DDLCATLOD) area</td>
</tr>
<tr>
<td><strong>dcdml</strong></td>
<td>Name of the system dictionary index (DDLCATX) area</td>
</tr>
<tr>
<td><strong>idms.system.dcdml</strong></td>
<td>ID of the system dictionary catalog index (DDLCATX) area</td>
</tr>
<tr>
<td><strong>dcdml</strong></td>
<td>Name of the system dictionary definition (DDLDML) area</td>
</tr>
</tbody>
</table>
```
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>idms.system.ddldml</td>
<td>ID of the system dictionary definition (DDLDML) area</td>
</tr>
<tr>
<td>dclod</td>
<td>Name of the system dictionary definition load (DDLDCLOD) area</td>
</tr>
<tr>
<td>idms.system.ddldcld</td>
<td>ID of the system dictionary definition load (DDLDCLOD) area</td>
</tr>
<tr>
<td>dclog</td>
<td>Name of the system log area (DDLDCLOG) area</td>
</tr>
<tr>
<td>idms.system.ddldclog</td>
<td>ID of the system log (DDLDCLOG) area</td>
</tr>
<tr>
<td>dcrun</td>
<td>Name of the system queue (DDLDCRUN) area</td>
</tr>
<tr>
<td>idms.system.ddldcrun</td>
<td>ID of the system queue (DDLDCRUN) area</td>
</tr>
<tr>
<td>dcsr</td>
<td>Name of the system scratch (DDLDCSCR) area</td>
</tr>
<tr>
<td>idms.system.ddldcsr</td>
<td>ID of the system scratch (DDLDCSCR) area</td>
</tr>
<tr>
<td>dcmsg</td>
<td>Name of the system message (DDLDCMSG) area</td>
</tr>
<tr>
<td>idms.sysmsg.ddldcmsg</td>
<td>ID of the system message (DDLDCMSG) area</td>
</tr>
<tr>
<td>dclscr</td>
<td>Name of the local mode system scratch (DDLOCSKR) area</td>
</tr>
<tr>
<td>idms.sysloc.ddlocsr</td>
<td>ID of the local mode system scratch (DDLOCSKR) area</td>
</tr>
<tr>
<td>dirldb</td>
<td>Name of the IDMSDIRL definition (DDLDML) area</td>
</tr>
<tr>
<td>idms.sysdir.ddldml</td>
<td>ID of the IDMSDIRL definition (DDLDML) area</td>
</tr>
<tr>
<td>dirlldl</td>
<td>Name of the IDMSDIRL definition load (DDLDCLOD) area</td>
</tr>
<tr>
<td>idms.sysdir.dirlldod</td>
<td>ID of the IDMSDIRL definition load (DDLDCLOD) area</td>
</tr>
<tr>
<td>empdemo</td>
<td>Name of the EMPDEMO area</td>
</tr>
<tr>
<td>idms.empdemo1</td>
<td>ID of the EMPDEMO area</td>
</tr>
<tr>
<td>insdemo</td>
<td>Name of the INSDEMO area</td>
</tr>
<tr>
<td>idms.insdemo1</td>
<td>ID of the INSDEMO area</td>
</tr>
<tr>
<td>orgdemo</td>
<td>Name of the ORGDEMO area</td>
</tr>
<tr>
<td>idms.orgdemo1</td>
<td>ID of the ORGDEMO area</td>
</tr>
<tr>
<td>empldem</td>
<td>Name of the EMPLDEMO area</td>
</tr>
<tr>
<td>idms.sqldemo.empldemo</td>
<td>ID of the EMPLDEMO area</td>
</tr>
<tr>
<td>infodem</td>
<td>Name of the INFXDEMO area</td>
</tr>
<tr>
<td>idms.sqldemo.infodemo</td>
<td>ID of the INFXDEMO area</td>
</tr>
<tr>
<td>projdem</td>
<td>Name of the PROJDEMO area</td>
</tr>
<tr>
<td>idms.projseg.projdemo</td>
<td>ID of the PROJDEMO area</td>
</tr>
<tr>
<td>indxdem</td>
<td>Name of the INDXDEMO area</td>
</tr>
<tr>
<td>idms.sqldemo.indxdem</td>
<td>ID of the INDXDEMO area</td>
</tr>
<tr>
<td>sysctl</td>
<td>Name of the SYSCTFL file</td>
</tr>
<tr>
<td>idms.sysctl</td>
<td>ID of the SYSCTFL file</td>
</tr>
<tr>
<td>secdd</td>
<td>Name of the system user catalog (DDLSEC) area</td>
</tr>
<tr>
<td>idms.sysuser.ddlsec</td>
<td>ID of the system user catalog (DDLSEC) area</td>
</tr>
<tr>
<td>dictdb</td>
<td>Name of the application dictionary definition area</td>
</tr>
<tr>
<td>idms.appldict.ddldml</td>
<td>ID of the application dictionary definition (DDLDML) area</td>
</tr>
</tbody>
</table>
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)

For more information, see the following topics:
- Beginning an OCF Session (see page 33)
- Conducting an Online Session (see page 34)
- Ending an OCF Session (see page 35)
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 (see page 11)).

4. Optionally, enter a SET OPTIONS statement to establish session-specific processing and output formatting options (see Using SET OPTIONS to Select Options (see page 12)).

OCF Screen Format

OCF uses a standard screen, which contains the following:

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

OCF uses a standard screen, which contains the following:

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

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 (see page 36)).

- Save the work file as a dictionary module (see Creating, Editing, and Saving Modules (see page 39)).

- Display the contents of any dictionary module for editing (see Creating, Editing, and Saving Modules (see page 39)).

**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 (see page 12)). 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 (see page 45)) or use the program function (PF) key assigned to the desired command function (see Using Control Keys (see page 66)).

Editing Options

You can use line-editing commands to edit the statements contained in your OCF work file (see Using Line Commands (see page 61)).

Coding Considerations

For more information on coding considerations, see Coding Considerations (see page 9).

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)

### Contents

- OCFX Status Codes (see page 38)

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**

```
OCFX module-name Version version-number Echo Noecho
```

**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).

\textit{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.

\textbf{Note:} For more information on the LANGUAGE parameter of the MODULE statement, see the \textit{CA IDMS IDD DDDL Reference Section}.

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

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

\textbf{Noecho}
Specifies that the named module's input statements are not to be displayed. If you use this parameter and also use \texttt{SET OPTIONS STATUS OFF} (see Using SET OPTIONS to Select Options (see page 12)) as the first statement in the module's source code, then the screen displays only non-zero status messages.

\textbf{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:

\texttt{PAGE \textit{nnn} \quad \textit{NEXT PAGE:}}

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

\begin{itemize}
  \item Press Enter or PA1 to page to the following screen.
  \item Press PA2 to page to the previous screen.
  \item Type a page number after the NEXT PAGE prompt to select a specific screen, then press Enter to page to that screen.
\end{itemize}

\textbf{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:

\begin{enumerate}
  \item Include the OCFX statement as one of the statements in the command list.
  \item Execute the command list.
\end{enumerate}
OCFX Status Codes

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

<table>
<thead>
<tr>
<th>Status code</th>
<th>Message and description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC394 001</td>
<td>INVALID SYNTAX TOKEN FOUND</td>
</tr>
<tr>
<td>DC394 002</td>
<td>INVALID MAJOR SCAN CODE &lt;nnn&gt;</td>
</tr>
<tr>
<td>DC394 003</td>
<td>COMMAND EXCEEDS AVAILABLE SPACE</td>
</tr>
<tr>
<td>DC394 004</td>
<td>INVALID DICTIONARY/NODE NAME</td>
</tr>
<tr>
<td>DC394 005</td>
<td>INTERNAL STORAGE EXCEEDED</td>
</tr>
<tr>
<td>DC394 006</td>
<td>MODULE NOT FOUND</td>
</tr>
<tr>
<td>DC394 007</td>
<td>MODULE HAS NO TEXT</td>
</tr>
<tr>
<td>DC394 008</td>
<td>UNEXPECTED DBMS ERROR</td>
</tr>
</tbody>
</table>

Note: For more information on other status codes, see the CA IDMS Messages and Codes Section.
Creating, Editing, and Saving Modules

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

Note: For more information on executing these modules, see Executing Statements Stored in Modules (OCFX) (see page 36).

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.

Using the Online Text Editor

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

Online compilers

Throughout the section, the term online compilers refers to the following CA IDMS development tools:
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:
Online Compilers

**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)
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

- CV node name -- Displays the CV node name.

Example

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

<table>
<thead>
<tr>
<th>Command area and release area</th>
<th>Compiler name</th>
<th>Message</th>
<th>Dictionary/ database</th>
<th>Current/ last line name</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDD nn.n</td>
<td>NO ERRORS</td>
<td>DICT=SYSDICT 1/29497 SYSTEM72</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

- HELP
- SUSPEND
- SWITCH

Note: For more information on top-line commands, see Using Top-line Commands (see page 45).

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)

Command area  Compiler name and release

Screen title  Screen name

IDD REL nn.n  *** RECORD ENTITY ***  RECD

RECORD 'EMPLOYEE' VERSION 100 DISPLAYED

X DISPLAY     RECORD NAME......: EMPLOYEE
- MODIFY
- ADD
- DELETE

VERSION NUMBER...: 100      _ HIGHEST      _ NEXT HIGHEST

_SYSTEM     _ LOWEST      _ NEXT LOWEST

DESCRIPTION......:

RECORD LENGTH....:

- RELM = RECORD ELEMENTS <PF9>
- RELL = REC ELEMENT LIST <PF10>
- REGN = USER REGISTRATION <PF2>
- CLAT = CLASS/ATTRIBUTES <PF4>
- COMM = COMMENTS <PF6>
- HIST = HISTORY <PF8>
- XREF = CROSS REFERENCE <PF8>

- COBL = COBOL ELEMENTS <PF11>
- RECX = RECORD EXTENSION <PF3>
- PUBL = PUBLIC ACCESS <PF5>
- COML = COMMENT KEY LIST <PF7>
- COPY = COPY FROM/SAME AS <PF7>
- 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.

```
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 (see page 40).

**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.

```
Module 'EMP-MOD' Version 1
-------1-------2-------3-------4-------5-------6-------7-------
OBTAIN EMP-JOB-LR WHERE DESCRIPTION-0440 = 'PROGRAMMER/ANALYST'.
```
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 on editing commands that you enter in the input/output area of the screen, see Using Line Commands (see page 61).

**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 (see page 66)).

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)

For more information, see the following topics:

- Summary of Top-line Commands (see page 46)
- APPLY (see page 47)
- CLEAR (see page 47)
- DELETE ALL (see page 47)
- DISPLAY LINE (see page 48)
- DISPLAY PAGE (see page 49)
- END (see page 50)
- ENTER (see page 50)
- ESCAPE (see page 50)
- FIND (see page 51)
- HELP (see page 51)
- INSERT (see page 52)
- PRINT (see page 54)
- REPEAT (see page 54)
- RESHOW (see page 54)
- SUSPEND (see page 55)
- SWAP (see page 55)
- SWITCH (see page 56)
- UPDATE (see page 61)
Summary of Top-line Commands

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

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLY</td>
<td>Updates the screen and work file but does not execute the statements in the OCF work file or execute the compiler.</td>
</tr>
<tr>
<td>CLEAR</td>
<td>Deletes all data contained in the work file.</td>
</tr>
<tr>
<td>DISPLAY LINE</td>
<td>Displays a page of the work file, starting with the specified line.</td>
</tr>
<tr>
<td>DISPLAY PAGE</td>
<td>Displays the requested page from the work file.</td>
</tr>
<tr>
<td>END</td>
<td>Immediately terminates the current session.</td>
</tr>
<tr>
<td>ENTER=APPLY</td>
<td>Sets the ENTER key to execute either the APPLY or the UPDATE command.</td>
</tr>
<tr>
<td>ENTER=UPDATE</td>
<td></td>
</tr>
<tr>
<td>ESCAPE</td>
<td>Establishes the escape character that must be used with line editing commands.</td>
</tr>
<tr>
<td>FIND</td>
<td>Locates a character string by searching forward or backward in the work file.</td>
</tr>
<tr>
<td>HELP</td>
<td>Lists each top-line command and the associated PF key currently assigned to execute that command.</td>
</tr>
<tr>
<td>INSERT</td>
<td>Inserts lines into the work file after the line at which the cursor is positioned.</td>
</tr>
<tr>
<td>PRINT</td>
<td>Prints the contents of the work file.</td>
</tr>
<tr>
<td>REPEAT</td>
<td>Repeats the line at which the cursor is positioned.</td>
</tr>
<tr>
<td>RESHOW</td>
<td>Cancels all changes made to the current screen and redispalyes the previous screen.</td>
</tr>
<tr>
<td>SUSPEND</td>
<td>Suspends the current session and returns control to the host TP monitor.</td>
</tr>
<tr>
<td>SWAP</td>
<td>Restores the screen and the work file to their condition prior to the last execution.</td>
</tr>
<tr>
<td>SWITCH (only if the development tool is executing under the control of the Transfer Control Facility)</td>
<td>Suspends the session and transfers control to the specified online CA IDMS component or to the Selection Screen of the Transfer Control Facility.</td>
</tr>
<tr>
<td>UPDATE</td>
<td>Updates the work file and executes the statements in the OCF work file or submits the work file to the compiler.</td>
</tr>
</tbody>
</table>

The rest of this section 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 PRI. 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

```plaintext
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

```plaintext
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 (see page 61).

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

```plaintext
EScape escape-character
```

Parameter

```plaintext
escape-character
```
• 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

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:
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

```plaintext
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

```plaintext
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 redispays 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
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 on the Transfer Control Facility, see Transfer Control Facility (see page 69).

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

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
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 on the TCF, see Transfer Control Facility (see page 69).

Syntax

```
SWITCH product-code task-code 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.

- **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.

  **Note:** The CLEar and CLR options are not used with ADSAT, ADSCT, or MAPCT task 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.
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.

<table>
<thead>
<tr>
<th>Development Tool</th>
<th>Product Code</th>
<th>Sample Task Code</th>
<th>Site Task Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA ADS application compiler (ADSA)</td>
<td>ADSA</td>
<td>ADSAT</td>
<td></td>
</tr>
<tr>
<td>CA ADS dialog compiler (ADSC)</td>
<td>ADSC</td>
<td>ADSCT</td>
<td></td>
</tr>
<tr>
<td>ASF</td>
<td>ASF</td>
<td>ASFT</td>
<td></td>
</tr>
<tr>
<td>IDD menu facility</td>
<td>IDDM</td>
<td>IDDMT</td>
<td></td>
</tr>
<tr>
<td>Online IDD</td>
<td>IDD</td>
<td>IDDT</td>
<td></td>
</tr>
<tr>
<td>Online map compiler (MAPC)</td>
<td>MAPC</td>
<td>MAPCT</td>
<td></td>
</tr>
<tr>
<td>Online command facility (OCF)</td>
<td>OCF</td>
<td>OCFT</td>
<td></td>
</tr>
<tr>
<td>CA OLQ</td>
<td>OLQ</td>
<td>OLQT</td>
<td></td>
</tr>
<tr>
<td>Online schema compiler</td>
<td>SCHEMA</td>
<td>SCHEMAT</td>
<td></td>
</tr>
<tr>
<td>Online subschema compiler</td>
<td>SSC</td>
<td>SSCT</td>
<td></td>
</tr>
<tr>
<td>Online system generation compiler</td>
<td>SYSGEN</td>
<td>SYSGENT</td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>SWITCH command specification</th>
<th>If you have suspended session(s)</th>
<th>If you do not have suspended session(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWITCH</td>
<td>Resumes most recently suspended</td>
<td>Displays TCF Selection Screen</td>
</tr>
<tr>
<td></td>
<td>development tool session</td>
<td></td>
</tr>
<tr>
<td>SWITCH prod</td>
<td>Resumes most recently suspended</td>
<td>Starts new session of requested tool</td>
</tr>
<tr>
<td>uct or task</td>
<td>session of requested tool (IDD</td>
<td>if tool is IDD or online compiler or</td>
</tr>
<tr>
<td>code OLD</td>
<td>and online compilers only)</td>
<td>displays DC/UCF screen if tool is ADSA,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ADSC, or MAPC</td>
</tr>
<tr>
<td>SWITCH prod</td>
<td>Resumes most recently suspended</td>
<td>Starts new session of requested tool,</td>
</tr>
<tr>
<td>uct or task</td>
<td>session of requested tool (IDD</td>
<td>if possible</td>
</tr>
<tr>
<td>code NEW</td>
<td>and online compilers only)</td>
<td></td>
</tr>
<tr>
<td>SWITCH tcf-task-code</td>
<td>Displays TCF Selection Screen</td>
<td>Displays TCF Selection Screen</td>
</tr>
<tr>
<td>SWITCH CLEAR</td>
<td>Clears list of suspended</td>
<td>Displays DC/UCF screen</td>
</tr>
<tr>
<td></td>
<td>development tool tool sessions</td>
<td></td>
</tr>
</tbody>
</table>
### 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.

<table>
<thead>
<tr>
<th>Development Tool</th>
<th>Specifying the SWITCH Command</th>
<th>Specifying Optional Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADSA, ADSC, MAPC</td>
<td>Select the SWITCH activity from the action bar of the main menu or Type SWITCH in the command line.</td>
<td>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</td>
</tr>
<tr>
<td>IDD</td>
<td>Type SWITCH in the command line.</td>
<td>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</td>
</tr>
<tr>
<td>OLQ</td>
<td>Type SWITCH in the command line.</td>
<td>Specify a task code as a SWITCH parameter Example: switch schema</td>
</tr>
<tr>
<td>Online schema, subschema, and system generation compilers, and the OCF</td>
<td>Type SWITCH in the command line.</td>
<td>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</td>
</tr>
</tbody>
</table>
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

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.

For more information, see the following topics:
- Copying Lines (see page 61)
- Deleting Lines (see page 62)
- Entering Block Commands (see page 63)
- Moving Lines (see page 64)
- Repeating Lines (see page 65)
- Repositioning Lines on the Screen (see page 66)

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:

<table>
<thead>
<tr>
<th>Use:</th>
<th>To:</th>
</tr>
</thead>
<tbody>
<tr>
<td>%C(space)</td>
<td>Copy the indicated line to the destination marked by the %A or %B command.</td>
</tr>
<tr>
<td>%Cn(space)</td>
<td>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.</td>
</tr>
</tbody>
</table>
| %CB(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.

```
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.

```
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
  EMP-NAME-0415-KBR
  %a USAGE IS DISPLAY.
```

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

```
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:**

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

---

### 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 This is the beginning of the text block
  This is the middle of the text block
  %me This is the end of the text block
  %a 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:

<table>
<thead>
<tr>
<th>Type of error</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mixing block commands</td>
<td></td>
</tr>
<tr>
<td>Type of error</td>
<td>Example</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>%CB (space) (copy begin)</td>
<td>. . .</td>
</tr>
<tr>
<td>. . .</td>
<td>%ME (space) (move end)</td>
</tr>
<tr>
<td>Specifying a target line inside a block</td>
<td>%CB (space)</td>
</tr>
<tr>
<td>. . .</td>
<td>%A (space) This is a target inside a block to be copied</td>
</tr>
<tr>
<td>. . .</td>
<td>%CE (space)</td>
</tr>
<tr>
<td>Specifying the start of the block after the end of the block</td>
<td>%CE (space) This is the end of a block . . .</td>
</tr>
<tr>
<td></td>
<td>%CB (space) This is the beginning of the block</td>
</tr>
</tbody>
</table>

### 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:
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.

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

The following sample screens illustrate the process of repeating lines. The following screen shows the

Repeating Lines

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

<table>
<thead>
<tr>
<th>Use</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>%R</td>
<td>Repeats the indicated line one time.</td>
</tr>
<tr>
<td>%Rn</td>
<td>Repeat the indicated line n times. For example, %R2 repeats the line two times.</td>
</tr>
<tr>
<td>%RBn</td>
<td>Repeat a block of lines n times. %RBn 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 topline command REPEAT. For example, %R100 is a valid command, REPEAT 100 is not.</td>
</tr>
</tbody>
</table>

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.

### 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.

For more information, see the following topics:
- Simulating PF Keys (see page 67)
- Using OCF and Online Compiler Control Keys (see page 68)
- Using IDD Menu Facility Global Control Keys (see page 68)
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
           PFKKEY 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)
```
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.

<table>
<thead>
<tr>
<th>Press (default)</th>
<th>Site key</th>
<th>To invoke</th>
<th>Which</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF1, PF13 PF8,</td>
<td></td>
<td>DISPLAY PAGE</td>
<td>Scrolls forward one page.</td>
</tr>
<tr>
<td>PF20</td>
<td></td>
<td>NEXT</td>
<td></td>
</tr>
<tr>
<td>PF2, PF14 PF7,</td>
<td></td>
<td>DISPLAY PAGE</td>
<td>Scrolls backward one page.</td>
</tr>
<tr>
<td>PF19</td>
<td></td>
<td>PRIOR</td>
<td></td>
</tr>
<tr>
<td>PF3, PF15</td>
<td></td>
<td>DISPLAY LINE</td>
<td>Scrolls forward one line.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NEXT</td>
<td></td>
</tr>
<tr>
<td>PF4, PF16</td>
<td></td>
<td>INSERT</td>
<td>Inserts up to a full screen of lines.</td>
</tr>
<tr>
<td>PF5, PF17</td>
<td></td>
<td>APPLY</td>
<td>Updates screen contents and work file but does not invoke the compiler.</td>
</tr>
<tr>
<td>PF6, PF18</td>
<td></td>
<td>UPDATE</td>
<td>Updates work file and executes the compiler.</td>
</tr>
<tr>
<td>PF9, PF21</td>
<td></td>
<td>SWAP</td>
<td>Restores the work file contents.</td>
</tr>
<tr>
<td>PF12, PF24</td>
<td></td>
<td>PRINT</td>
<td>Prints the work file contents.</td>
</tr>
<tr>
<td>PA1</td>
<td></td>
<td>Cancel FIND</td>
<td>Cancels the FIND command.</td>
</tr>
<tr>
<td>PA2</td>
<td></td>
<td>RESHOW</td>
<td>Cancels changes to the current screen and redisplays the screen.</td>
</tr>
<tr>
<td>Clear</td>
<td></td>
<td>CLEAR</td>
<td>Clears the contents of the work file.</td>
</tr>
<tr>
<td>Enter</td>
<td></td>
<td>=APPLY =UPDATE</td>
<td>Updates the screen and the work file.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Updates the work file and executes the compiler.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Press</th>
<th>To invoke</th>
<th>Which</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF1, PF13</td>
<td>HELP</td>
<td>Displays online help tutorial associated with the current screen.</td>
</tr>
</tbody>
</table>
Using Transfer Control Facility (TCF)

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).

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.

For more information, see the following topics:
- Tools You Can Invoke from TCF (see page 69)
- Examples: Transfer Control Facility (see page 70)
- Basics of Using TCF (see page 75)
- TCF Selection Screen (see page 81)
- Development Tools under TCF (see page 85)
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 on how development tools are established under TCF, see the *CA IDMS System Generation Section*.

**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 on the SWITCH command, see SWITCH (see page 56).

---

**Examples: Transfer Control Facility**

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
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CA ADS Online Dialog Compiler</td>
</tr>
<tr>
<td>CA, Inc.</td>
</tr>
</tbody>
</table>
```

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) <yyyy> CA. All rights reserved. ADSC 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 . . . . . . . . . . . . . . . . EMPPMAP
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) <yyyy> CA. All rights reserved.
MAPC is currently running under control of TCF

Command ===>
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
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. 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.
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)
- TERMINATE TCF SESSION (PF3)

*TCP TASKCODES*

SELECT ONE TO START A NEW SESSION

- TCF
- ADSCT DIALOG COMPILER
- ADSAT APPLICATION COMPILER MAPC
- ASF AUTOMATIC SYSTEM FACILITY
- IDDMT IDD MENU MODE
- IDDT IDD COMMAND MODE
- MAPCT MAP COMPILER
- OCFT ONLINE COMMAND FACILITY
- OLOT CA OLQ COMMAND MODE
- SCHEMAT SCHEMA COMPILER
- SSCT SUBSCHEMA COMPILER
- SYSGENT SYSGEN COMPILER

*SUSPENDED SESSIONS*

SELECT ONE TO RESUME AN OLD SESSION

- ADSCT MODEMP 0001
- MAPC EMPMAP 0001
- IDD

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.

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
Basics of 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. 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.

For more information, see the following topics:
- Using the TCF (see page 75)
- Transferring Between Development Tools (see page 78)
- Invoking a Development Tool (see page 78)
- Exiting from a Development Tool (see page 79)
- Exiting from TCF (see page 80)

Using the TCF

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 on using specific tools under TCF, see Development Tools under TCF (see page 85).

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 on 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 on transferring from specific tools under TCF, see Development Tools under TCF (see page 85).

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.
<table>
<thead>
<tr>
<th>Development tool</th>
<th>TCF response when you request a new development tool session</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADSA</td>
<td>Because you can have two or more simultaneous sessions of each of these tools, TCF starts a new session of the requested tool.</td>
</tr>
<tr>
<td>ADSC</td>
<td>Because you can have two or more simultaneous sessions of each of these tools, TCF starts a new session of the requested tool.</td>
</tr>
<tr>
<td>MAPC</td>
<td>Because you can have two or more simultaneous sessions of each of these tools, TCF starts a new session of the requested tool.</td>
</tr>
<tr>
<td>IDD menu facility</td>
<td>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.</td>
</tr>
<tr>
<td>Online command facility</td>
<td>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.</td>
</tr>
<tr>
<td>Online IDD</td>
<td>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.</td>
</tr>
<tr>
<td>Online schema compiler</td>
<td>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.</td>
</tr>
<tr>
<td>Online system generation compiler</td>
<td>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.</td>
</tr>
<tr>
<td>ASF</td>
<td>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.</td>
</tr>
<tr>
<td>CA OLQ</td>
<td>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.</td>
</tr>
</tbody>
</table>

**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 (see page 81).

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 on the transfer of control from a specific development tool, see the discussion of that development tool in Development Tools under TCF (see page 85).

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 (see page 80)).
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 on the Selection Screen and your list of suspended sessions, see TCF Selection Screen section (see page 81).

**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 section (see page 81).
- 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.
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) _ TERMINATE TCF SESSION (PF3)

*TCF TASKCODES*
SELECT ONE TO START A NEW SESSION

<table>
<thead>
<tr>
<th>TASKCODE</th>
<th>DESCRIPTOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCF</td>
<td></td>
</tr>
<tr>
<td>ADSCT</td>
<td>DIALOG COMPILER</td>
</tr>
<tr>
<td>ADSAT</td>
<td>APPLICATION COMPILER</td>
</tr>
<tr>
<td>ASFT</td>
<td>AUTOMATIC SYSTEM FACILITY</td>
</tr>
<tr>
<td>IDDMT</td>
<td>IDD MENU MODE</td>
</tr>
<tr>
<td>IDDT</td>
<td>IDD COMMAND MODE</td>
</tr>
<tr>
<td>MAPCT</td>
<td>MAP DEFINITION</td>
</tr>
<tr>
<td>OCFT</td>
<td>ONLINE COMMAND FACILITY</td>
</tr>
<tr>
<td>OLQT</td>
<td>OLQ COMMAND MODE</td>
</tr>
<tr>
<td>SCHEMAT</td>
<td>SCHEMA COMPILER</td>
</tr>
<tr>
<td>SSCT</td>
<td>SUBSCHEMA COMPILER</td>
</tr>
<tr>
<td>SYSGENT</td>
<td>SYSGEN COMPILER</td>
</tr>
</tbody>
</table>

*SUSPENDED SESSIONS*
SELECT ONE TO RESUME AN OLD SESSION

<table>
<thead>
<tr>
<th>TASKCODE</th>
<th>DESCRIPTOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADSAT</td>
<td>MODEMP 001</td>
</tr>
<tr>
<td>ADSCT</td>
<td>EMPINFO 001</td>
</tr>
<tr>
<td>ASFT</td>
<td>EMPLIST 0001</td>
</tr>
<tr>
<td>IDDMT</td>
<td></td>
</tr>
<tr>
<td>IDDT</td>
<td></td>
</tr>
</tbody>
</table>
```

**TCF Selection Screen Options**
<table>
<thead>
<tr>
<th>Option</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exit from TCF</td>
<td>SUSPEND TCF SESSION (PF9)</td>
<td>Returns control to DC/UCF. Saves your list of suspended development tool sessions.</td>
</tr>
<tr>
<td></td>
<td>TERMINATE TCF SESSION (PF3)</td>
<td>Returns control to DC/UCF. Clears your list of suspended development tool sessions.</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Specify a database or dictionary name</th>
<th>DBNAME</th>
<th>Names the database (if any) that you have most recently specified.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DBNODE</td>
<td>Names the database node (if any) that you have most recently specified.</td>
</tr>
<tr>
<td></td>
<td>DICTNAME</td>
<td>Names the dictionary (if any) that you have most recently specified.</td>
</tr>
<tr>
<td></td>
<td>DICTNODE</td>
<td>Names the dictionary node (if any) that you have most recently specified.</td>
</tr>
</tbody>
</table>

**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)

<table>
<thead>
<tr>
<th>Start a new development tool session</th>
<th>TCF</th>
<th>Task code for TCF Selection Screen</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ADSCT</td>
<td>Task code for ADSC</td>
</tr>
<tr>
<td></td>
<td>ADSAT</td>
<td>Task code for ADSA</td>
</tr>
<tr>
<td></td>
<td>ASFT</td>
<td>Task code for ASF</td>
</tr>
<tr>
<td>Option</td>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>IDDMT</td>
<td></td>
<td>Task code for IDD menu facility</td>
</tr>
<tr>
<td>IDDT</td>
<td></td>
<td>Task code for online IDD</td>
</tr>
<tr>
<td>MAPCT</td>
<td></td>
<td>Task code for MAPC</td>
</tr>
<tr>
<td>OCFT</td>
<td></td>
<td>Task code for online command facility</td>
</tr>
<tr>
<td>OLQT</td>
<td></td>
<td>Task code for OLQ</td>
</tr>
<tr>
<td>SCHEMAT</td>
<td></td>
<td>Task code for online schema compiler</td>
</tr>
<tr>
<td>SSCT</td>
<td></td>
<td>Task code for online subschema compiler</td>
</tr>
<tr>
<td>SYSGENT</td>
<td></td>
<td>Task code for system generation compiler</td>
</tr>
</tbody>
</table>

**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
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RESUME</td>
<td>List 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.</td>
</tr>
<tr>
<td></td>
<td>TASKCODE</td>
<td>Lists the task code for each suspended development tool session.</td>
</tr>
<tr>
<td></td>
<td>DESCRIPTOR</td>
<td>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.</td>
</tr>
</tbody>
</table>

**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.
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 section.

- CA ADS and the Online Map Compiler (see page 85)
- CA IDMS Online Compilers (see page 87)
- CA OLQ under TCF (see page 89)
- ASF (see page 90)
- IDD (see page 91)

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 the following 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 below.

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 displays. 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) <yyyy> CA. All rights reserved.
Command ===>
Enter  F1=Help  F3=Exit  F10=Action

Using a Program Function Key

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
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. 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.

<table>
<thead>
<tr>
<th>Exit Method</th>
<th>TCF Actions</th>
</tr>
</thead>
</table>
| 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 (see page 56). |
| 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.

<table>
<thead>
<tr>
<th>Exit Method</th>
<th>TCF Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUSPEND command</td>
<td>Suspends your compiler session. Transfers control to your most recently</td>
</tr>
<tr>
<td></td>
<td>suspended development tool session or to the Selection screen if you have</td>
</tr>
<tr>
<td></td>
<td>no sessions.</td>
</tr>
<tr>
<td>SWITCH command</td>
<td>Suspends your compiler session. Transfers control to the requested</td>
</tr>
<tr>
<td></td>
<td>development tool. If you do not request a tool, control transfers as if you</td>
</tr>
<tr>
<td></td>
<td>used SUSPEND (see the description for SUSPEND command (see page 55)).</td>
</tr>
<tr>
<td></td>
<td>For more information on SWITCH, see SWITCH (see page 56).</td>
</tr>
</tbody>
</table>
Exit Method | TCF Actions
---|---
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 under TCF

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-NAMES-0415 :
EMP-FIRST-NAMES-0415 : ROGER
EMP-LAST-NAMES-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

Select Command/ Screen Name Help
Pfkey Option Description
--- 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=BACKWARD

---
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.

<table>
<thead>
<tr>
<th>Exit Method</th>
<th>TCF Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUSPEND command (command mode only)</td>
<td>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.</td>
</tr>
<tr>
<td>SWITCH command</td>
<td>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 on SWITCH, see SWITCH (see page 56).</td>
</tr>
<tr>
<td>BYE, QUIT, or SIGNOFF command (or associated function key)</td>
<td>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.</td>
</tr>
</tbody>
</table>

(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
- PF2 - Define Table
- PF3 - Load Data
- PF4 - Display/Change Data
- PF5 - Select Data
- PF7 - Page Backward
- PF8 - Page Forward
- PF9 - Passkey
- PF11 - Change
- PF12 - Query
- PF13 - Query
- PF14 - Signon
- PA1 - Prior Level

Table Name:
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.

<table>
<thead>
<tr>
<th>Exit Method</th>
<th>TCF Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF13-- Query</td>
<td>Transfers control to CA OLQ, saving signon and table information.</td>
</tr>
<tr>
<td>CLEAR-- Leave ASF</td>
<td>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.</td>
</tr>
</tbody>
</table>

### 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
```

15-Jan-2018 91/102
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.

<table>
<thead>
<tr>
<th>Exit Method</th>
<th>TCF Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUSPEND command</td>
<td>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.</td>
</tr>
<tr>
<td>SWITCH command</td>
<td>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 on SWITCH, see SWITCH (see page 56).</td>
</tr>
<tr>
<td>BYE, END, LOGOFF, or SIGNOFF  command or CLEAR key, as appropriate</td>
<td></td>
</tr>
</tbody>
</table>
Using the Command Facility as a Subprogram

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.

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

For more information, see the following topics:

- Command Facility Subprogram Basics (see page 94)
- Compiler Interface Parameters (see page 94)
- Work-area File (see page 97)
- Sample Program that Calls the Command Facility (see page 97)
Command Facility Subprogram Basics

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 (see page 97).

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 Parameters

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:

<table>
<thead>
<tr>
<th>Field Label</th>
<th>Usage Size</th>
<th>Picture Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CIOID</td>
<td>Char 4 X(4) Compiler I/O ID; must be initialized to 'CIO'</td>
</tr>
<tr>
<td>2</td>
<td>CIOUSER</td>
<td>4</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 (X'04')</td>
<td>An invalid parameter has been passed to the compiler interface; typically, the CIO or CIOF block has not been properly initialized.</td>
</tr>
<tr>
<td>8 (X'08')</td>
<td>The scratch id or queue id specified in the CIOF block could not be found.</td>
</tr>
<tr>
<td>28 (X'1C')</td>
<td>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.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Field</th>
<th>Label</th>
<th>Usage</th>
<th>Size</th>
<th>Picture</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIOF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SZM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Fields 1 and 2

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

<table>
<thead>
<tr>
<th>File type</th>
<th>File name</th>
</tr>
</thead>
<tbody>
<tr>
<td>For a work area initialize to 'WORKAREA'</td>
<td>Not applicable</td>
</tr>
<tr>
<td>For a scratch area initialize to 'SCRATCH '</td>
<td>Scratch area ID</td>
</tr>
<tr>
<td>For a queue initialize to 'QUEUE '</td>
<td>Queue ID</td>
</tr>
<tr>
<td>For a program initialize to 'LINKPGM ' or 'LINKEPA', based on the LINK mechanism used</td>
<td>If 'LINKPGM' use the 8-character program name&lt;br&gt; If 'LINKEPA' use the 4-character entry point address</td>
</tr>
<tr>
<td>For a null file initialize to 'NULL'</td>
<td>Not applicable</td>
</tr>
</tbody>
</table>

### 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:

<table>
<thead>
<tr>
<th>Parameter Information passed</th>
<th>Information passed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 CIO block</td>
<td></td>
</tr>
<tr>
<td>2 CIOF block for input file</td>
<td></td>
</tr>
<tr>
<td>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'</td>
<td></td>
</tr>
<tr>
<td>4 CIOF block for the output print file</td>
<td></td>
</tr>
<tr>
<td>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'</td>
<td></td>
</tr>
<tr>
<td>6 CIOF block for the output punch file</td>
<td></td>
</tr>
<tr>
<td>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'</td>
<td></td>
</tr>
</tbody>
</table>
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.

```cobol
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-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.
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.*

**IDMS AREA**

COPY IDMS SUBSCHEMA-CTRL.
COPY IDMS MAP-CONTROLS.
COPY IDMS MAP-RECORDS.
EJECT
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.

0800-RETURN-SCREEN.
  ACCEPT TASK CODE INTO TASK-CODE.
  DC RETURN NEXT TASK CODE TASK-CODE.

0900-DC-RETURN.
  DELETE SCRATCH RECORD ID SCR-RCDID.
  DC RETURN.
  EJECT

1000-GET-SCRATCH-REC.
  MOVE 1 TO SCR-RCDID.
  GET SCRATCH RECORD ID SCR-RCDID KEEP
  INTO SCR-RCR TO SCR-RCR-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.

2000-DISPLAY-MAP.
  MOVE 0 TO SCR-DBK.
  MOVE '1' TO SCR-STATUS.
  PUT SCRATCH FROM SCR-RCR TO SCR-RCR-END
  RECORD ID SCR-RCDID.
  PERFORM 8000-INITIALIZE-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-INITIALIZE-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-I0-RC NOT = GOOD-RC
      MOVE CIO-I0-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.
5999-EXIT.
EXIT.
******************************************************************************
* 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.