HP System Dictionary/XL Cobol Definition Extractor Reference Manual

HP 3000 MPE/iX Computer Systems

Edition 1

Manufacturing Part Number: 32257-90001
E1287

U.S.A. December 1987

Notice

The information contained in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for direct, indirect, special, incidental or consequential damages in connection with the furnishing or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

Restricted Rights Legend

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013. Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19 (c) (1,2).

Acknowledgments

Hewlett-Packard Company
3000 Hanover Street
Palo Alto, CA 94304 U.S.A.

Preface

Manual Organization

This manual is the standard reference manual for users of HP System Dictionary/XL COBOL Definition Extractor which runs on the 900 Series HP 3000 computer system. It is a reference document for all persons developing COBOL II/XL source code on the 900 Series HP 3000. As such, it assumes both a working knowledge of the 900 Series HP 3000 and the COBOL II/XL programming language.

This manual is organized as follows:

Chapter 1 COBOL Definition Extractor Presents an overview of the HP System Dictionary/XL COBOL Definition Extractor.

Chapter 2 Running the SDCDE Program Describes the files that the HP System Dictionary/XL COBOL Definition Extractor uses, the run options, error handling, and other information you may need when executing the utility.

Chapter 3 SDCDE Commands Describes the syntax and parameters of the HP System Dictionary/XL COBOL Definition Extractor commands.

Chapter 4 Source Generation Describes the source generation process.

Chapter 5 Definition Translation Describes the translation of System Dictionary definitions into COBOL source code.

Appendix A SDCDE Error Messages Provides a listing of the HP System Dictionary/XL error messages that are returned on a System Dictionary COBOL Definition Extractor error, one or more probable causes of each error, and an action for each listed cause that corrects the problem.

Appendix B SDCDE Command Abbreviations Provides a table listing the HP System Dictionary/XL COBOL Definition Extractor commands and their abbreviation.

Appendix C SDCDE PICTURE Clause Translation Describes the translation of PICTURE clauses.

Glossary Provides a glossary of the major terms associated with HP System Dictionary.

Audience

The intended users of this manual are those individuals developing COBOL II source code on the HP 3000 computer. Knowledge of the HP 3000 MPE XL operating system is helpful.

Resources

In addition to this manual, you may need to consult the following manuals:

Managing Your Information Network: A Data Dictionary Primer

HP System Dictionary/XL SDMAIN Reference Manual

HP System Dictionary/XL Intrinsics Reference Manual

HP System Dictionary/XL Utilities Reference Manual

HP System Dictionary/XL General Reference Manual, Volume 1

HP System Dictionary/XL General Reference Manual, Volume 2

HP System Dictionary/XL COBOL Definition Extractor Reference Manual

TurboIMAGE/XL Reference Manual

SQL Reference Manual

HP SQL Database Administration Guide

VPLUS Reference Manual

QUERY/V Reference Manual

KSAM Reference Manual

HP 3000 General Information Manual

MPE XL Commands Reference Manual

MPE XL Intrinsics Reference Manual

Program Design and Optimization

Programmer's Utilities and Tools

Native Language Support Reference Manual

EDIT/V Reference Manual

Pascal/XL Reference Manual

Pascal/XL Programmer's Guide

COBOL II/XL Reference Manual

COBOL II/XL Programmer's Guide

HP FORTRAN 77 Reference Manual

HP FORTRAN 77/XL Reference Manual Supplement

HP FORTRAN 77/XL Programmer's Guide

HP FORTRAN 77/XL Programmer's Guide Supplement

SPL Reference Manual

Future Upgrades

In the future, Hewlett-Packard will prefix all names of objects it adds to the core set with "HP" . To avoid potential name conflicts, do not create any entity types, relationship classes, attributes, scopes, domains, or

versions prefixed with "HP" .

Conventions

NOTATION **DESCRIPTION**

nonitalics — Words in syntax statements which are not in italics must be entered exactly as shown. Punctuation characters other than brackets, braces and ellipses must also be entered exactly as shown. For example: EXIT;

italics — Words in syntax statements which are in italics denote a parameter which must be replaced by a user-supplied variable. For example: CLOSE *filename*

[ ] — An element inside brackets in a syntax statement is optional. Several elements stacked inside brackets means the user may select any one or none of these elements. For example:

[A]

[B] User *may* select A or B or neither.

{ } — When several elements are stacked within braces in a syntax statement, the user must select one of those elements. For example:

{A}

{B} User *must* select A or B or C.

{C}

... — A horizontal ellipsis in a syntax statement indicates that a previous element may be repeated. For example: [,*itemname*]...;

In addition, vertical and horizontal ellipses may be used in examples to indicate that portions of the example have been omitted.

, — A shaded delimiter preceding a parameter in a syntax statement indicates that the delimiter *must* be supplied whenever (a) that parameter is included or (b) that parameter is omitted and any *other* parameter which follows is included. For example:

itema [, *itemb*][, *itemc*]

means that the following are allowed:

itema

itema,itemb

itema,itemb,itemc

itema,,itemc

Å — When necessary for clarity, the symbol Å may be used in a syntax statement to indicate a required blank or an exact number of blanks. For example: SET[(*modifier*)] Å (*variable*);

underlining — Brackets, braces or ellipses appearing in syntax or format statements which must be entered as shown will be underlined. For example:

LET *var* [[*subscript*]] = *value*

Output and input/output parameters are underlined. A notation in the description of each pa-

rameter distinguishes input/output from output parameters. For example:

CREATE (*parm1*, parm2, flags, error)

shading Shading represents inverse video on the terminal's screen. In addition, it is used to emphasize key portions of an example.

[[ ]] The symbol [[ ]] may be used to indicate a key on the terminal's keyboard. For example, [[RETURN]] indicates the carriage return key.

[[CONTROL]]*char* Control characters are indicated by [[CONTROL]] followed by the character. For example, [[CONTROL]]Y means the user presses the control key and the character Y simultaneously.

COBOL Definition Extractor Utility

Product Overview

The HP System Dictionary/XL COBOL Definition Extractor Utility generates COBOL II source code into copy libraries from definitions residing in System Dictionary. This manual describes the operation of this utility.

- Chapter 1 describes the files used by SDCDE, the run options, error handling, and other information you may need when executing the utility.
- Chapter 2 describes the syntax and parameters of SDCDE commands.
- Chapter 3 describes the source generation process.
- Chapter 4 describes the translation of System Dictionary definitions into COBOL source.
- Appendix A lists all of the SDCDE error messages.
- Appendix B lists the SDCDE command abbreviations.
- Appendix C describes the translation of PICTURE clauses.
- A glossary of the major terms associated with HP System Dictionary.

1 Running the SDCDE Program

Running SDCDE

The SDCDE program is located in the file SDCDE.PUB.SYS. You can load and run the SDCDE program by issuing the command:

```
RUN SDCDE.PUB.SYS
```

You do not need to specify any file equations. If the program loads successfully, SDCDE prints out the following banner:

```
HP System Dictionary SDCDE HP32257v.uu.ff - (C)Hewlett-Packard Co.  1985
```

SDCDE Files

The SDCDE program uses an **input file** and a **log file**. These two files are discussed below.

The Input File

SDCDE accepts input from the file, SDIN. SDCDE can also take input from another MPE file if you redirect SDIN with an MPE file equation prior to issuing the RUN command:

```
:FILE SDIN=input file
```

If SDIN does not exist, or has not been redirected to another file, $STDINX (or your terminal) is used as the input file.

The Log File

When SDCDE is executed, all valid commands are logged to the log file with the formal file designator, SDLOG. This enables you to save the commands and use them at a later date.

The REDO command and any invalid commands are not logged. However, commands edited and executed through the REDO command are logged. Any responses to prompts (either valid or invalid) that appear only in session mode are omitted from the log file, for example, the prompt that asks permission to overwrite SDLOG.

When you run SDCDE in session mode, if SDLOG already exists, SDCDE asks for permission to overwrite the old log file. If the response is negative, logging is disabled. On the other hand, in batch mode, SDLOG is automatically overwritten. If no old log file is found and no file equation for SDLOG is found, SDCDE creates the file. This file is temporary, however, and must be saved to become a permanent file. If SDLOG is new, you receive a message telling you to save the file upon termination of the program.

You can save and rename the log file and then use it as the input file (SDIN) the next time you run SDCDE.

SDCDE Run Options

When you run SDCDE, you can specify one of several run options through the use of the PARM parameter of the RUN command. This parameter allows you to control SDCDE's error handling. Using this parameter, you can specify the exact number of errors at which the program aborts. If, for example, two errors are acceptable, but more than two errors are not, then specify the number 3 on the PARM parameter. An example of the RUN command with the PARM parameter appears below:

```
RUN SDCDE.PUB.SYS; PARM=3
```

Three sets of values are accepted for the run option:

-1	Parse-only option. Only command syntax is checked. No dictionary or copy library is opened.
0	Execute and do not terminate regardless of the number of errors. Program continues until an EXIT command is issued. All errors are reported, but non-fatal errors do not terminate the program. This is the default option.
n	Terminate when n errors are detected. Syntactical, operational, and illegal response errors are counted. If a fatal error is encountered, the program may abort before the error count is reached. Positive parm values are not recognized in session mode.

NOTE When the program aborts due to too many errors, an error message is issued, and the program sets the system Job Control Word (JCW) to a value greater than or equal to FATAL.

Error Handling

SDCDE recognizes four different categories of errors. SDCDE responds to these errors in different ways, depending upon the category. The four SDCDE error categories are described below.

Syntax errors are errors caused either by mistyping a command or by specifying an illegal command construction. When these errors are detected, a caret pointer is displayed pointing to the location of the error, along with an error message.

Operational warnings are errors that can occur only during source generation, but are not considered to be serious enough to require termination of the process. When such an error occurs, a warning message is issued and source generation continues.

Operational errors are caused by commands that cannot start or continue execution, due to an invalid operation. For example, issuing the GENERATE command when no dictionary is open is an operational error. Another example occurs when an error is encountered during source generation that is considered serious enough to terminate the process. An error message is displayed in response to these errors.

Fatal errors are usually disc-related errors over which the program has very little or no control. When such an error is detected, the program is terminated. Failure to open $STDINX is such an error. When a fatal error is detected by the system, the Job Control Word (JCW) is assigned a value greater than or equal to FATAL.

Control Y Handling

You can issue a [[CONTROL]] Y at any time during the operation of SDCDE to return to the command prompt (>). When you issue [[CONTROL]] Y, SDCDE displays:

```
< Control Y >
```

Session Mode vs. Batch Mode You can run SDCDE in either session mode or in batch mode. Some differences exist, however, between the two modes. When running in batch mode, both [[CONTROL]] Y and the REDO command are disabled. Also note that, in batch mode, any old SDLOG file is automatically overwritten. No confirmation prompt is issued. When running in session mode, any run option other than 0 or -1 is ignored.

2 SDCDE Commands

Introduction

This section includes descriptions of all SDCDE commands, their syntax, parameters, and examples using the commands. These commands are listed alphabetically.

A HELP command is provided for a quick reference to SDCDE commands. Using this command, you can either get a list of all available commands, or a detailed description of a particular command.

The abbreviation of each command and subcommand is listed in Appendix B.

MPE Commands

You can issue MPE commands without leaving SDCDE. Whenever the command prompt (>) is present, you can issue an MPE command by entering a colon (:) followed by the desired command. SDCDE passes the command on to the operating system to be processed. After the command is processed, control returns to the user with the command prompt (>).

You can extend an MPE command over multiple lines by attaching the ampersand (&) continuation character to the end of the line. However, unlike other SDCDE multi-line commands, you should not attach the ending period.

The MPE commands that you can issue from SDCDE are the ones that are supported by the MPE COMMAND intrinsic. Refer to the *MPE/XL Intrinsics Reference Manual* for a list of these commands.

COBEDIT Command

The COBEDIT command uses the COBEDIT utility to edit and examine COBOL copy libraries. The currently opened copy library is closed prior to entering the COBEDIT utility. Upon termination of COBEDIT, the copy library is reopened.

Syntax

```
COBEDIT [.]
```

Example

The following is an example using the COBEDIT command.

```
>COBEDIT
Closing current copy library
HP32233A.00.15 COPYLIB EDITOR - COBEDIT THU, JUN 5, 1986,  3:43 PM
(C) HEWLETT-PACKARD CO. 1985
TYPE "HELP" FOR A LIST OF COMMANDS.
>
.
.
.
>E
```

```
COBEDIT run terminated.  Resuming SDCDE
Old copy library opened
>
```

COMMENT Command

The COMMENT command allows you to enter description text into the command stream. It allows the use of text to document what is happening in the job stream or SDIN file without having any effect on the operation of the utility. You can use the COMMENT command whenever the command prompt (>) is present.

The command is a very simple one. It consists of the command word COMMENT followed by a blank and then zero or more characters. The program ignores all characters on the line and simply writes them to the log file for documentation. The program then prompts for the next command. Note that the COMMENT command does not need to end in a period and so the command must be contained on a single line. You can specify multi-line comments by prefacing each line with the COMMENT command.

Syntax

```
COMMENT  [.]
```

Example

The following example adds five lines of comments to the command stream and log file.

```
>COMMENT
>COMMENT   This command file creates a copy
>COMMENT   library from the System Dictionary
>COMMENT   for the payroll system.
>COMMENT
>
```

COPYLIB Command

The COPYLIB command opens the COBOL copy library into which the COBOL source is generated. Through this command you can name an old copy library or create a new one. When specifying a new copy library, you must include the KEY-FILE parameter.

Syntax

```
COPYLIB [;NAME = file-name ]
        [;KEY-FILE = kfile-name ]
        [;UPDATE-DICT = update-option ]
        [;OPEN-MODE = open-mode ]
        {.}
```

Parameters

file-name — Name of the copy library KSAM file. If the named KSAM file does not exist, SDCDE assumes that the file is new, and uses the KEY-FILE parameter value to create a new file. There is no default for this parameter.

kfile-name — Name of the key file of a new copy library. If the named copy library already exists, the value of this parameter is ignored. There is no default for this parameter.

update-option Through this option, you can enter the entity/relationship definitions involving the copy library and the key file into System Dictionary. When UPDATE-DICT = ON, and a new copy library is created, the COPYLIB entity definition and the KSAMFILE entity definition are created from the copy library and key file name, respectively. Then the COPYLIB key KSAMFILE relationship is created with the entity definitions. Note that in order create these definitions, the dictionary must be opened in either shared-update or exclusive-update open mode and the STATUS parameter of the DEFINE command must be TEST. SDCDE issues an error message if you create a copy library when the UPDATE-DICT parameter is set to ON and the dictionary is not open or open in the wrong mode. When this happens SDCDE creates the copy library, but does not update the dictionary. You can set the UPDATE-DICT parameter without opening a copy library. The initial default for this parameter is OFF.

open-mode When generating COBOL source, if the module name you specify already exists in the copy library, you can either have the new source replace the lines that already exist in the module or you can have the new source appended to the end of the old lines. Specify REPLACE if you want to replace an existing module in the copy library. Because of KSAM, the REPLACE option does not delete copylib records, it flags them as deleted. Therefore it is possible to hit an end of file in the copy library even when REPLACE = ON. Specify APPEND if you want to append to an existing module in the copy library. You can set the OPEN-MODE parameter without opening a copy library. The initial default for this parameter is APPEND.

Example

The following example creates a new copy library.

```
>COPYLIB NAME=CL1001
>>        KEY-FILE=CL1001K
>>        UPDATE-DICT=OFF
>>        OPEN-MODE=REPLACE.
New copy library created
>
```

DEFINE Command

The DEFINE command opens the System Dictionary. You must open the dictionary before you can generate COBOL source with the GENERATE command. You can generate COBOL source code with the STANDARDS-PARM command without opening the dictionary. When opening the dictionary for the first time, the scope-name and scope-password are required.

Syntax

```
DEFINE [;DICTIONARY = dictionary-filename]
       [;SCOPE = scope-name]
       [;PASSWORD = [scope-password]]
       [;DOMAIN = [domain-name]]
       [;VERSION = [version-name]]
       [;STATUS = version-status]
       [;OPEN-MODE = open-mode]
       [;NAME-MODE = name-mode]
```

{.}

Parameters

dictionary-filename Specifies the name of the dictionary to be opened. The initial default for the DICTIONARY parameter is SYSDIC. After the dictionary is opened, the default is the currently opened dictionary.

scope-name Specifies the name of the scope from which to retrieve definitions. The SCOPE parameter is required when opening the dictionary for the first time.

scope-password Gives access to the scope. Any characters are allowed in a password. If a character in the password is not valid in other System Dictionary names (that is, on the restricted list), you must enter the password within quotes to allow recognition of the "restricted" characters.

If you specify the SCOPE parameter and not the PASSWORD parameter, you will be prompted for the password. For security reasons, the echo is turned off and you are given three chances to enter the correct password. If you are in session mode and the correct password is not entered after three tries, the DEFINE command terminates and the command prompt (>) is displayed. If you are in batch mode and the correct password is not entered after three tries, the program terminates. The password is always read from $STDINX.

domain-name Specifies the name of the dictionary domain in which the dictionary is to be opened. The initial default for the DOMAIN parameter is the common domain. After the dictionary is opened, the default is the current domain. If you are in another domain and you want to get back to the common domain, specify "DOMAIN=;" with no value.

version-name Specifies the name of the version under which the dictionary is opened. When the VERSION parameter is blank, the value of the STATUS parameter opens the latest version of the current status. The default version is blank.

version-status The value of the STATUS parameter opens the latest version of this status. Options are:

TEST Uses the version status TEST. Dictionary definitions in a version of TEST status can be modified.

PRODUCTION Uses the version status PRODUCTION. Dictionary definitions in a version of PRODUCTION status can only be read.

ARCHIVAL Uses the version status ARCHIVAL. Dictionary definitions in a version of ARCHIVAL status can only be read.

The STATUS parameter is effective only when the value of the VERSION parameter is blank. If you specify both the VERSION parameter and the STATUS parameter with the DEFINE command, only the VERSION parameter is used. SDCDE issues a warning message indicating that the STATUS parameter was ignored. The initial default for the STATUS parameter is PRODUCTION. After the dictionary is open, the default is the current status value.

open-mode Specifies the mode in which to open the dictionary. Options are:

READ-ALLOW-READ Allows you to read System Dictionary data definitions, but does not allow definitions to be created when the UPDATE-DICT option of the COPYLIB command is ON. Others can access the dictionary only for reading.

READ-ONLY Allows you to read System Dictionary data definitions, but does not allow definitions to be created when the UPDATE-DICT option of the COPYLIB command is ON. Others can also access the dictionary.

SHARED-UPDATE Allows you to read definitions and create definitions when the UPDATE-DICT option of the COPYLIB command is ON. Others can also access the dictionary.

EXCLUSIVE- UPDATE Allows you to read definitions and create definitions when the UPDATE-DICT option of the COPYLIB command is ON. No one else can access the dictionary.

The initial default is to open the dictionary in SHARED-UPDATE mode.

name-mode Specifies which group of names (internal or external) you wish to use. Options are:

INTERNAL Uses internal names. Internal names can never change.

EXTERNAL Uses external names. External names are fully customizable and localizable.

The default for the first open is EXTERNAL.

Example

The following example opens the dictionary SYSDIC using the scope MANAGER in SHARED-UPDATE open mode. The password MGR* allows access to the scope.

```
>DEFINE DICTIONARY=SYSDIC;
>>SCOPE=MANAGER;
>>PASSWORD=MGR*;
>>OPEN-MODE=SHARED-UPDATE.
>
```

EXIT Command

The EXIT command terminates the SDCDE program. The EXIT command closes the dictionary, copy library, and other opened files before terminating the program.

Syntax

```
EXIT  [.]
```

Example

The following example exits the program (closing the dictionary and any files used by the system) and returns to the operating system prompt.

```
>EXIT.
 END OF PROGRAM
 :(Back to MPE)
```

GENERATE Command

The GENERATE command starts the COBOL source generation process as described in Chapter 4. The dictionary and COBOL copy library must be open.

Syntax

```
GENERATE {entity-name}
         {;ENTITY-TYPE = entity-type}
         {.}
```

Parameters

entity-name — Name of the entity for which source generation occurs. This is the name that is used in the generated code, unless you specify an alias attribute through the ALIAS parameter of the OPTIONS command. If you specify an alias attribute, SDCDE checks for that alias attribute, and if one exists, uses the alias attribute value in place of the name.

entity-type — Entity type of the specified entity. Options are:

IMAGE-DATABASE

IMAGE-DATASET

FORMSFILE

FORM

FILE

KSAMFILE

RECORD

ELEMENT

HELP Command

A HELP command is available for a quick reference of SDCDE commands. You can use the HELP command either to get a list of all available SDCDE commands or to get a detailed description of a specific command.

Syntax

```
HELP  [command] [.]
```

Parameter

command — Name of the command for which you are requesting help.

Description

If you request more HELP messages than can fit on one screen, the following message is displayed:

```
Continue (Y/N) >
```

Respond "Y" or [[Return]] if you wish to see more messages, "N" if you do not.

If the HELP command you specify contains a subject with no HELP message available, an error message is issued, and the general HELP messages are displayed.

Example

The following example shows the text that your screen displays when you specify the HELP command.

```
>HELP.
```

```
MAIN HELP SCREEN
Commands accepted by SDCDE:
COBEDIT         <CE  > - Run the COBEDIT subsystem
COMMENT         <COM > - Provide a line of comment text in the command stream
COPYLIB         <CL  > - Establish the COBOL COPYLIB environment
DEFINE          <DEF > - Establish the System Dictionary environment
EXIT            <E   > - Terminate the program
GENERATE        <GEN > - Generate COBOL source for a System Dictionary entity
HELP            <H   > - Display commands
LIST            <L   > - List occurrences of a specified entity-type
OPTIONS         <OP  > - Set parameters for the source generation process
REDO            <REDO> - Edit and execute the previous command
RESET           <R   > - Reset all the DEFINE, COPYLIB, and OPTIONS parameters
SHOW            <SH  > - Display the current values of DEFINE, COPYLIB, and
                         OPTIONS parameters
STANDARD-PARMS <SP  > - Generate standard parameters for IMAGE, KSAM, or VPLUS
>
```

LIST Command

The LIST command lists all of the entities of the specified type contained in the System Dictionary that is currently open, or lists all of the modules contained in the copy library that is currently open. The number of occurrences or modules retrieved is printed at the end of the occurrence list or module list.

If you specify the LIST command without an argument, HELP messages for the LIST command appear.

Syntax

```
     [entity-type]
LIST [MODULE     ] [.]
```

Parameters

entity-type — Specifies the entity type of the entities to be listed. You can specify the following entity types:

- COPYLIB
- IMAGE-DATABASE
- IMAGE-DATASET
- FORMSFILE
- FORM
- FILE
- KSAMFILE
- RECORD
- ELEMENT

MODULE — Lists all of the modules contained in the copy library that is currently open.

OPTIONS Command

The OPTIONS command alters values of the options that directly affect the source generation process. The options remain in effect in the current session until you override them with another OPTIONS command or until you reset them with the RESET command.

Syntax

```
OPTIONS [;ALIAS = alias-type]
        [;EDIT-MASK = mask-option]
        [;QUALIFY = qualify-option]
        [;ELEM-QUALIFY = elem-qual-type]
        [;CONSTANTS = constants-option]
        [;SELECT-FILE = select-file-option]
        [;COMMON-STORE = common-store-option]
        [;SECONDARY-REC = secondary-rec-option]
        [;ECHO = echo-option]
        [;COMMENT = comment-option]
        {.}
```

Parameters

alias-type — Specifies a keyword that the System Dictionary uses to select an alias attribute that SDCDE checks during the source generation process. The following keywords are available:

NONE	SDCDE uses the entity name.
COBOL	SDCDE checks the cobol-alias attribute.
IMAGE	SDCDE checks the image-alias attribute.
STANDARD	SDCDE checks the standard-alias attribute.
VPLUS	SDCDE checks the vplus-alias attribute.

The initial default is NONE. When you select a keyword other than NONE, SDCDE checks the corresponding attribute and if it is non-blank, uses its value in place of the entity name. For example, if you specify ALIAS = COBOL, SDCDE checks the cobol-alias attribute and if it is not blank, uses its attribute value in place of the entity name. Refer to Chapter 4, "Source Generation" , for details on how SDCDE generates aliases.

mask-option — Specifies how the edit mask of the PICTURE clause is generated for elements. You can specify ON or OFF. If you specify EDIT-MASK = ON, SDCDE uses the edit-mask attribute to generate the PICTURE clause. If you specify EDIT-MASK = OFF, SDCDE uses the element-type, byte-length, decimal, and display-length attributes to generate the edit mask of the PICTURE clause. Note the following about the edit-mask attribute:

If you use the edit-mask attribute, all other attributes pertaining to the PICTURE clause are ignored. However, if the data type (whether numeric or alphanumeric) cannot be determined from the edit-mask attribute alone, the element-type attribute is used to determine it.

If the edit-mask attribute is blank or non-existent, the PICTURE clause for that element is generated as if EDIT-MASK = OFF. The element's data length and display length are both determined from the edit-mask. Hence, the byte-offset attribute should be consistent with the

edit-mask, not the byte-length or display-length attributes.

The edit-mask attribute can either be in the COBOL edit mask format or the Dictionary/3000 edit mask format. If the edit-mask contains inconsistent or erroneous data, a warning message is issued and the edit-mask is not used. Instead, the edit mask is generated as if EDIT-MASK = OFF.

See Appendix C for the mapping of the edit-mask attribute to the COBOL edit mask and the computation of the edit mask from the element-type, byte-length, display-length and decimal attributes.

qualify-option Allows qualification of all non-element names with their parent names. You can specify ON or OFF. If you specify QUALIFY = ON, all non-element entity names are qualified with the parent name. If the combined name is longer than 61 characters, SDCDE truncates the name. If you specify QUALIFY = OFF, the parent name qualification is suppressed. The default is OFF.

elem-qual-type Specifies prompting for element prefixes or suffixes. The following options are available:

PREFIX	During the source generation process, SDCDE asks for element prefixes.
SUFFIX	During the source generation process, SDCDE asks for element suffixes.
NONE	During the source generation process, no prompts for element qualification are issued.

The default is PREFIX.

constants-option Allows constants related to an entity to be generated. For an IMAGE-DATABASE entity type, the constants generated include the database name, database passwords, data set names, and their search items. For a FORMSFILE entity type, the constants generated include the forms file name, forms file lockword, head form name, all form names, and each form's field number table. You can specify ON or OFF. If you specify CONSTANTS = ON, constants are generated. If you specify CONSTANTS = OFF, constants are not generated. The default is ON.

select-file-option Allows prompting for SELECT and FILE section module names when generating COBOL source code for FILE and KSAMFILE entity types. You can specify ON or OFF. If you specify SELECT-FILE = ON, SDCDE prompts you for SELECT and FILE section module names during the source generation process of the FILE and KSAMFILE entity types. If you specify SELECT-FILE = OFF, COBOL source is generated only into the WORKING-STORAGE SECTION and not into the FILE or SELECT sections. The default is ON.

common-store-option Allows all records and forms contained in one entity to use the same memory area. This is useful if space conservation is an important issue for your program. You can specify ON or OFF. If you specify COMMON-STORE = ON, you can have all the records or forms contained in one IMAGE-DATABASE or FORMSFILE use the same memory area. SDCDE does this by generating REDEFINES clause. For example, suppose you have a database DB that contains data sets DS-1 and DS-2. If COMMON-STORE = ON, the following COBOL source is generated:

```
01 DS-1-DATA.
   ...
01 DS-2-DATA REDEFINES DS-1-DATA.
```

```
...
```

If you specify COMMON-STORE = OFF, then different records and forms occupy different memory areas. For example, suppose again that you have a database DB that contains data sets DS-1 and DS-2. If COMMON-STORE = OFF, the following COBOL source is generated:

```
01 DS-1-DATA.
   ...
01 DS-2-DATA.
   ...
```

The default is OFF.

secondary-rec-option Allows COBOL source generation for secondary records as well as primary records. Entities like IMAGE-DATASET, FILE, and KSAMFILE can have more than one record contained in them. Only one should be designated as the primary record, and the rest secondary. If you specify SECONDARY-REC = ON, SDCDE generates the primary record first and then the secondary records are generated with the REDEFINES clause. If you specify SECONDARY-REC = OFF, SDCDE generates only the primary record definition. The default is OFF.

echo-option Allows suppression of the echoing of the generated source into $STDLIST. You can specify ON or OFF. If you specify ECHO = ON, the generated source is echoed in $STDLIST. If you specify ECHO = OFF, the generated source is not echoed. The default is ON.

comment- option Allows the COBOL source that is generated to be preceded by comments that describe the entity. You can specify ON or OFF. If you specify COMMENT = ON, the generated COBOL source is preceded by comments that describe the entity. If an element is generated, only the original element is commented. If any entity other than elements are generated, all entities contained in that entity are commented, except for elements. For example, if an IMAGE-DATABASE entity is generated, the database, data sets, and records are all commented, but not the elements. If you specify COMMENT = OFF, the generated COBOL source is not preceded by comments. The default is OFF. See Chapter 5 for the list of attributes that are included as comments for each entity type.

REDO Commands

The REDO command allows you to correct errors or to make changes to the last command you issued. When you issue the REDO command, you enter an editing mode and the first line of the command is displayed for modification.

Syntax

```
REDO [.]
```

Subcommands

To modify the command, position the cursor using the space bar, under the character to be modified and enter one of the subcommands listed below.

SUBCOMMAND	DESCRIPTION
A	Appends one or more characters to the end of the current line, regardless of the position of the command.

SUBCOMMAND	DESCRIPTION
B	Breaks the line into two lines, moving the character that is above the cursor and all following characters to the next line. The second line becomes the current edit line.
D	Deletes the character above the D. If you repeat D , each character above each D is deleted. You may also use a D below the first and last character to be deleted with spaces in between.
E	Exits the REDO editing mode without executing the edited command. The command that you were editing when you entered E is still considered to be the last one.
H	Lists all available editing sub-commands in REDO mode. Your current editing line is then redisplayed.
I	Inserts one or more characters immediately preceding the character that is above the cursor. You can combine a DELETE and INSERT edit by using D's followed by an I and the characters to be inserted.
L	Lists the complete command as it is currently edited and then redisplays the line you are currently editing.
R	Replaces the characters above the cursor with the new characters you enter. The first character to be replaced is the one above the R.
X	Executes the current command as it has been edited.
+n	Places the cursor on the next line of the command you are editing. You can enter + followed by a number n, for the number of lines you want to skip forward. If you do not enter a number, the default is to move forward one line.
-n	Places the cursor on the previous line of the command you are editing. You can enter - followed by a number n, for the number of lines you want to skip backward. If you do not enter a number, the default is to is to move backward one line.
[[RETURN]]	Places the cursor on the next line of the command you are editing. If you are on the last line, the command executes as it has been edited.

Description

To modify the REDO command, use the subcommands just described. If you enter any character other than a valid subcommand, that character and the following characters are interpreted as replacement characters and they replace the characters above them in the command. For example, if you type TYPE below a set of characters, the word TYPE replaces the four characters in the command line above.

Once you issue the REDO command, you are placed in edit mode and the first line of the command is displayed for modification. If the line you want to edit is the second line of the command, rather than the first, the + subcommand can move the cursor forward to the desired line. The - subcommand can move the cursor backward to a previous line. The + or - subcommands can also be followed by a number, allowing the cursor to move forward or backward more than one line at a time.

Example

The following example corrects the typographical error for the word "scope" (which is misspelled "scopee") and then executes the corrected command.

```
>DEFINE DICTIONARY = SYSDIC;
>>      SCOPEE = MANAGER;
>>      PASSWORD = MGR*;
>>      OPEN-MODE = SHARED-UPDATE
     SCOPEE = MANAGER;
    ^
Unknown DEFINE parameter (SDERR 4534)
Text from the error to the end of command ignored (SDWARN 4521)
>REDO
DEFINE DICTIONARY = SYSDIC;


    SCOPEE = MANAGER;
          D
    SCOPE = MANAGER;
X
>
```

RESET Command

The RESET command closes the System Dictionary and copy library, and sets all option parameters to the default values.

Syntax

```
RESET [.]
```

Example

The following example shows the value of each parameter after the RESET command has been issued. You can obtain this list by issuing the SHOW command. If a parameter does not have an existing default value and you have not specifically assigned a value to the parameter, asterisks (***) are displayed. A blank value for DOMAIN indicates the common domain. A blank value for VERSION indicates the latest version of current status.

```
>RESET
>SHOW
DEFINE COMMAND PARAMETERS
  DICTIONARY       = SYSDIC
  SCOPE          = ***
  DOMAIN          =
  VERSION         =
  STATUS         = PRODUCTION
```

```
  OPEN-MODE        = SHARED-UPDATE
  NAME-MODE        = EXTERNAL
COPYLIB COMMAND PARAMETERS
  NAME             = ***
  KEY-FILE         =
  UPDATE-DICT      = OFF
  OPEN-MODE        = APPEND
OPTIONS COMMAND PARAMETERS
  ALIAS            = NONE
  EDIT-MASK        = OFF
  QUALIFY          = OFF
  ELEM-QUALIFY     = PREFIX
  CONSTANTS        = ON
  SELECT-FILE      = ON
  COMMON-STORE     = OFF
  SECONDARY-REC    = OFF
  ECHO             = ON
  COMMENT          = OFF
```

SHOW Command

The SHOW command displays the values of all SDCDE parameters. Default values are displayed for those values that are not specifically defined. If a parameter does not have an existing default value and you have not specifically assigned a value to the parameter, asterisks (***) are displayed.

Syntax

```
SHOW [.]
```

Example

The following example displays the values of all SDCDE parameters.

```
>SHOW
DEFINE COMMAND PARAMETERS
  DICTIONARY       = SYSDIC
  SCOPE            = MANAGER
  DOMAIN           =
  VERSION          = VERSION-ONE
  STATUS           = PRODUCTION
  OPEN-MODE        = SHARED-UPDATE
```

```
    NAME-MODE        = EXTERNAL

  COPYLIB COMMAND PARAMETERS
    NAME             = LIB1000
    KEY-FILE         = LIB1000K
    UPDATE-DICT      = OFF
    OPEN-MODE        = APPEND

  OPTIONS COMMAND PARAMETERS
    ALIAS            = COBOL
    EDIT-MASK        = OFF
    QUALIFY          = OFF
    ELEM-QUALIFY     = PREFIX
    CONSTANTS        = ON
    SELECT-FILE      = ON
    COMMON-STORE     = OFF
    SECONDARY-REC    = OFF
    ECHO             = ON
    COMMENT          = OFF
```

STANDARD-PARMS Command

The STANDARD-PARMS command allows generation of certain standard parameters for the IMAGE, KSAM, and VPLUS subsystems. You do not need to open the System Dictionary to use the STANDARD-PARMS command, but a copy library must be open.

Syntax

```
STANDARD-PARMS [subsys-name]
        {.}
```

Parameters

subsys-name — Name of the subsystem for which standard parameters are to be generated. The following subsystems are allowable:

IMAGE	Generates IMAGE subsystem standard parameters.
KSAM	Generates KSAM subsystem standard parameters.
VPLUS	Generates VPLUS subsystem standard parameters.

When you issue the STANDARD-PARMS command, you are prompted for the module name into which the standard parameters are to be generated. The prompt is one of the following, depending on the subsystem you specify:

Module for IMAGE standard parameters >

Module for KSAM standard parameters >

Module for VPLUS standard parameters >

If you issue STANDARD-PARMS without a subsystem name, you are prompted for module names for all three subsystems.

IMAGE Subsystem Standard Parameters

If you request the IMAGE standard parameters to be generated, the status array, the eight IMAGE intrinsic modes, and the intrinsic data item list declarations are generated into the specified module in the WORKING-STORAGE SECTION as shown below:

```
000100
000200 01  STATUSS.
000300     05  C-WORD          PIC S9(4) COMP.
000400     05  STAT2           PIC S9(4) COMP.
000500     05  STAT3-4         PIC S9(9) COMP.
000600     05  STAT5-6         PIC S9(9) COMP.
000700     05  STAT7-8         PIC S9(9) COMP.
000800     05  STAT9-10        PIC S9(9) COMP.
000900
001000
001100 01  EMPTY-LIST          PIC X(2) VALUE "; ".
001200 01  ALL-ITEMS           PIC X(2) VALUE "@;".
001300 01  PREVIOUS-LIST       PIC X(2) VALUE "*;".
001400
001500
001600 01  MODE1               PIC 9999 COMP VALUE 1.
001700 01  MODE2               PIC 9999 COMP VALUE 2.
001800 01  MODE3               PIC 9999 COMP VALUE 3.
001900 01  MODE4               PIC 9999 COMP VALUE 4.
002000 01  MODE5               PIC 9999 COMP VALUE 5.
002100 01  MODE6               PIC 9999 COMP VALUE 6.
002200 01  MODE7               PIC 9999 COMP VALUE 7.
002300 01  MODE8               PIC 9999 COMP VALUE 8.
```

VPLUS Subsystem Standard Parameters

If you requested generation of VPLUS standard parameters, the VPLUS comarea definition and other constants are generated as shown below:

```
000100
000200 01  COMAREA.
000300     05  COM-STATUS       PIC S9(4)  COMP  VALUE 0.
000400     05  COM-LANGUAGE     PIC S9(4)  COMP  VALUE 0.
000500     05  COM-COMAREALEN   PIC S9(4)  COMP  VALUE 60.
000600     05  FILLER           PIC S9(4)  COMP  VALUE 0.
000700     05  COM-MODE         PIC S9(4)  COMP  VALUE 0.
000800     05  COM-LASTKEY      PIC S9(4)  COMP  VALUE 0.
000900     05  COM-NUMERRS      PIC S9(4)  COMP  VALUE 0.
001000     05  FILLER           PIC S9(4)  COMP  VALUE 0.
001100     05  FILLER           PIC S9(4)  COMP  VALUE 0.
001200     05  FILLER           PIC S9(4)  COMP  VALUE 0.
001300     05  COM-CFNAME       PIC X(15)  VALUE SPACES.
001400     05  FILLER           PIC X(1)  VALUE SPACES.
001500     05  COM-NFNAME       PIC X(15)  VALUE SPACES.
001600     05  FILLER           PIC X(1)  VALUE SPACES.
001700     05  COM-REPEATOPT    PIC S9(4)  COMP  VALUE 0.
001800     05  COM-NFOPT        PIC S9(4)  COMP  VALUE 0.
001900     05  FILLER           PIC S9(4)  COMP  VALUE 0.
002000     05  COM-DBUFLEN      PIC S9(4)  COMP  VALUE 0.
002100     05  FILLER           PIC S9(4)  COMP  VALUE 0.
002200     05  FILLER           PIC S9(4)  COMP  VALUE 0.
002300     05  COM-DELETEFLAG   PIC S9(4)  COMP  VALUE 0.
002400     05  COM-SHOWCONTROL  PIC S9(4)  COMP  VALUE 0.
002500     05  FILLER           PIC S9(4)  COMP  VALUE 0.
002600     05  FILLER           PIC S9(4)  COMP  VALUE 0.
002700     05  FILLER           PIC S9(4)  COMP  VALUE 0.
002800     05  FILLER           PIC S9(4)  COMP  VALUE 0.
002900     05  FILLER           PIC S9(4)  COMP  VALUE 0.
003000     05  FILLER           PIC S9(4)  COMP  VALUE 0.
003100     05  FILLER           PIC S9(4)  COMP  VALUE 0.
003200     05  FILLER           PIC S9(4)  COMP  VALUE 0.
003300     05  COM-NUMRECS      PIC S9(9)  COMP  VALUE 0.
003400     05  COM-RECNUM       PIC S9(9)  COMP  VALUE 0.
```

```
003500    05 FILLER          PIC S9(4) COMP VALUE 0.
003600    05 FILLER          PIC S9(4) COMP VALUE 0.
003700    05 COM-TERMFILENUM  PIC S9(4) COMP VALUE 0.
003800    05 FILLER          PIC S9(4) COMP VALUE 0.
003900    05 FILLER          PIC S9(4) COMP VALUE 0.
004000    05 FILLER          PIC S9(4) COMP VALUE 0.
004100    05 FILLER          PIC S9(4) COMP VALUE 0.
004200    05 FILLER          PIC S9(4) COMP VALUE 0.
004300    05 FILLER          PIC S9(4) COMP VALUE 0.
004400    05 COM-TERMOPTIONS  PIC S9(4) COMP VALUE 0.
004500    05 FILLER          PIC S9(4) COMP VALUE 0.
004600    05 FILLER          PIC S9(4) COMP VALUE 0.
004700    05 FILLER          PIC S9(4) COMP VALUE 0.
004800    05 FILLER          PIC S9(4) COMP VALUE 0.
004900
005000
005100 01 TERMFILENAME       PIC X(6) VALUE "TERM ".
005200 01 MESSAGE-BUF        PIC X(72) VALUE SPACES.
005300 01 MESSAGE-BUF-LEN    PIC S9(4) COMP VALUE 72.
005400 01 MSGLEN             PIC S9(4) COMP VALUE 0.
005500 01 FIELDNUM           PIC S9(4) COMP VALUE 0.
005600 01 BUFLEN             PIC S9(4) COMP VALUE 0.
```

KSAM Subsystem Standard Parameters

If you request KSAM subsystem standard parameters to be generated, a general KSAM file table that does not reference any particular file and the KSAM status parameter are generated as follows:

```
000100
000200 01 FILETABLE.
000300    05 FILENUMBER      PIC S9(4) COMP VALUE 0.
000400    05 FILENAME        PIC X(8) VALUE SPACES.
000500    05 I-O-TYPE        PIC S9(4) COMP VALUE 0.
000600    05 A-MODE          PIC S9(4) COMP VALUE 0.
000700    05 PREV-OP         PIC S9(4) COMP VALUE 0.
000800
000900
```

```
001000 01  STAT.
001100     05  STATUS-KEY-1     PIC X.
001200     05  STATUS-KEY-2     PIC X.
```

3 Source Generation

Overview

You can request COBOL source generation by specifying the GENERATE command. When you specify this command, SDCDE verifies that the given entity exists for the given entity type and then prompts you for additional information before starting the source generation process.

Each entity type requires different information for the source generation process. The following sections describe the prompts that SDCDE issues for the valid entity types.

Suppose we have entities ELEMENT-1, ELEMENT-2, ... ELEMENT-N that are related as follows:

ELEMENT-1 contains ELEMENT-2

ELEMENT-2 contains ELEMENT-3

ELEMENT-3 contains ELEMENT-4

.

.

.

ELEMENT-N-1 contains ELEMENT-N

When COBOL source generation completes for ELEMENT-1, the COBOL record looks like the following:

```
01 ELEMENT-1
 05 ELEMENT-2
   10 ELEMENT-3
     15  ELEMENT-4
          .
          .
          .
       40  ELEMENT-N-1
         45  ELEMENT-N
```

For each iteration of the contains relationship, the level number increases by 5.

Creating COBOL Names

SDCDE automatically prefixes or suffixes some COBOL data names. This is in addition to the user-controlled prefixes, suffixes, and qualifications. SDCDE does this for two reasons:

- Prefixes help identify the data item's usage. For example, because the "DB-" prefix is always attached to the constants for database names, you can easily identify constants of this kind.
- Sometimes one entity may generate a number of COBOL data names of different usage. For example, a data set generates a COBOL record as well as a constant for the data set name. In this case, a prefix or suffix is necessary to create two distinct names.

Table 4-1 shows the prefixes and suffixes that SDCDE attaches to different kinds of COBOL data names. Note that record names are suffixed, and the constants are prefixed.

If an illegal COBOL character is found in an entity name, it is replaced with a "-." Then, if the first or the last character of the name is a "-," it is prefixed or suffixed with an "X-" or a "-X." If an entity name is a COBOL reserved word, it is suffixed with an "-X" . When any of these conditions occurs, SDCDE issues warning messages.

The maximum length for COBOL names in SDCDE is 61 characters. If a name is longer than 61 characters, it is truncated.

Table 1: COBOL Prefixes and Suffixes

Entity Type	COBOL Data Type	Prefix/ Suffix	Prefix/ Suffix Name
IMAGE-DATABASE	Constant for DB name Constant for DB password	Prefix Prefix	DB- PWD-
IMAGE-DATASET	Record name Constant for DS name Constant for search item	Suffix Prefix Prefix	-DATA DS- DI-
FORMSFILE	Constant for FF name Constant for FF lockword Constant for head form name	Prefix Prefix Prefix	FF- LWD- HF-
FORM	Record name Constant for FORM name Record name for field #'s Constant for field #	Suffix Prefix Suffix Prefix	-DATA FORM- -FIELDS FIELDNO-
FILE	Record name Record name in FILE section Constant for file name Constant for file lockword	Suffix Suffix Prefix Prefix	-DATA -REC FILE- LWD-
KSAMFILE	Record name Record name in FILE section Constant for KSAM file name Constant for file lockword File table constant for KSAM	Suffix Suffix Prefix Prefix Suffix	-DATA -REC KSAM- LWD- -FILETAB

Table 1: COBOL Prefixes and Suffixes

Entity Type	COBOL Data Type	Prefix/ Suffix	Prefix/ Suffix Name
RECORD	Record name	Suffix	-DATA
ELEMENT	Record name (If ELEMENT contain another ELEMENT)	Suffix	-ELEM
HP-CONDITION-NAME	Condition name constant	Prefix	CN-

Responses to SDCDE Prompts

At any SDCDE prompt, you can enter one of the following special responses:

?	Displays HELP messages associated with the current prompt.
]]	Returns to the command prompt.
]	Moves you back to the previous prompt. If there is no previous prompt, then the current prompt is reissued.
@	Causes SDCDE to respond as if you have entered[[RETURN]] at all prompts following the current one.
^Y	Equivalent to "]]" at prompts.

Alias Attribute Values

System Dictionary allows you to assign alias attribute values to any entity or relationship. SDCDE recognizes a limited set of alias attributes (cobol-alias, image-alias, standard-alias, and vplus-alias). You can select one of these aliases with the ALIAS parameter of the OPTIONS command. SDCDE only recognizes these aliases when you assign them to entities of the types listed in Table 4-1 or assign them to relationships of the type RECORD contains ELEMENT or ELEMENT contains ELEMENT. When you assign an alias to an entity, SDCDE interprets the alias as an alternate name for the entity, and uses the name in place of the entity name in the generated source code. When you assign an alias to a relationship, SDCDE interprets it as an alternate name for the second entity in the relationship. If a given entity has two values for the same alias attribute, one at the entity level and one at the relationship level, the relationship level attribute takes precedence.

For example, suppose you have the relationship CUSTOMER-NAME contains LAST-NAME of type ELEMENT contains ELEMENT. The LAST-NAME entity has a cobol-alias value of LAST-NAME-ITEM, and the relationship has a cobol-alias value of CUSTOMER-LAST-NAME. If the ALIAS parameter of the OPTIONS command is set to COBOL, then SDCDE generates the CUSTOMER-NAME item as follows:

```
01  CUSTOMER-NAME
    05  CUSTOMER-LAST-NAME
```

Note that SDCDE uses the relationship level alias rather than the entity level alias or the entity name.

Generating IMAGE-DATABASE Definitions

If you specify IMAGE-DATABASE as the entity type in the GENERATE command, a complete set of

COBOL record definitions is generated for the data sets contained in the given database. To determine whether to generate all data sets and constants in one module, or in separate modules, SDCDE issues the following prompt:

Define all data sets in one module (Y/N) >

If you enter a Y, SDCDE generates all definitions into one module. If you respond with an N, SDCDE generates each data set definition and the constants into different modules.

Using a Single Module

When you ask for a single module, SDCDE asks for the module name with the following prompt:

Module name for IMAGE-DATABASE *data-base-name* >

Enter the module name into which you wish to generate all data set and constant definitions. If the module already exists, the copy library's OPEN-MODE parameter determines whether to append all generated code to the existing module or to replace it. If you enter a [[RETURN]], SDCDE does not generate any code for this database and the GENERATE command will be complete.

If you specify the ELEM-QUALIFY parameter of the OPTIONS command, SDCDE responds with one of the following prompts, depending on the value you specify:

Value	Prompt
PREFIX	Element prefix >
SUFFIX	Element suffix >
NONE	No prompt

SDCDE attaches the prefix or suffix characters to the elements contained by the data sets. If you enter a [[RETURN]], SDCDE does not attach a prefix or suffix.

When answers to these prompts are complete, SDCDE starts the source generation process. The database definition is created for the WORKING-STORAGE SECTION, with each data set record corresponding to an 01 level COBOL record, and each element contained in the data set record corresponding to the 05 level element. (See "Generating IMAGE-DATASET definitions" for further explanations regarding data set generation.)

If you set the QUALIFY parameter of the OPTIONS command to ON, then SDCDE prefixes the COBOL record name for each data set with the database name.

If you set the CONSTANTS parameter of the OPTIONS command to ON, SDCDE generates the database name and password constants after the record layouts. SDCDE then generates the data set names and search item constants. The constant values generated use the image-alias attribute value if one exists, and the primary name if no image-alias attribute value is found. This rule applies regardless of the value of the ALIAS parameter of the OPTIONS command.

If you set the COMMON-STORE parameter of the OPTIONS command to ON, all data sets in the database occupy the same memory space via the REDEFINES clause.

Example

In the following **single module** example the database name is STORE, ELEM-QUALIFY = PREFIX, and CONSTANT = ON.

```
> GENERATE STORE; ENTITY-TYPE=IMAGE-DATABASE.
```

```
Define all data sets in one module (Y/N) > Y
Module name for IMAGE-DATABASE STORE > MODULE
Element prefix > [[RETURN]]
000100
000200 01  CUSTOMER-DATA.
000300     05  ACCOUNT              PIC S9(9)  COMP.
000400     05  LAST-NAME            PIC X(16).
000500     05  FIRST-NAME           PIC X(10).
000600     05  MID-INITIAL          PIC X(2).
000700     05  STREET-ADDRESS       PIC X(26).
000800     05  CITY                 PIC X(12).
000900     05  STATE                PIC X(2).
001000     05  ZIP                  PIC X(6).
001100     05  CREDIT-RATING        PIC S9(9)  COMP.
001200
001300
001400 01  DATE-MASTER-DATA.
001500     05  DATE-INFO            PIC X(6).
001600
001700
001800 01  PRODUCT-DATA.
001900     05  STOCK-NO             PIC X(8).
002000     05  DESCRIPTION          PIC X(20).
002100
002200
002300 01  SALES-DATA.
002400     05  ACCOUNT              PIC S9(9)  COMP.
002500     05  STOCK-NO             PIC X(8).
002600     05  QUANTITY             PIC S9(4)  COMP.
002700     05  PRICE                PIC S9(9)  COMP.
002800     05  TAX                  PIC S9(9)  COMP.
002900     05  TOTAL                PIC S9(9)  COMP.
003000     05  PURCH-DATE           PIC X(6).
003100     05  DELIV-DATE           PIC X(6).
```

```
003200
003300
003400 01  SUP-MASTER-DATA.
003500     05  SUPPLIER            PIC X(16).
003600     05  STREET-ADDRESS      PIC X(26).
003700     05  CITY                PIC X(12).
003800     05  STATE               PIC X(2).
003900     05  ZIP                 PIC X(6).
004000
004100
004200 01  INVENTORY-DATA.
004300     05  STOCK-NO            PIC X(8).
004400     05  ONHANDQTY           PIC S9(9)  COMP.
004500     05  SUPPLIER            PIC X(16).
004600     05  UNIT-COST           PIC S9(7)  COMP-3.
004700     05  LASTSHIPDATE        PIC X(6).
004800     05  BINNUM              PIC S9(2).
004900
005000
005100 01  DB-STORE                PIC X(8)  VALUE "  STORE ".
005200 01  PWD-STORE               PIC X(10)  VALUE "PASSWORD ".
005300
005400 01  DS-CUSTOMER             PIC X(9)  VALUE "CUSTOMER ".
005500 01  DS-DATE-MASTER          PIC X(12)  VALUE "DATE-MASTER ".
005600 01  DS-PRODUCT              PIC X(8)  VALUE "PRODUCT ".
005700 01  DS-SALES                PIC X(6)  VALUE "SALES ".
005800 01  DS-SUP-MASTER           PIC X(11)  VALUE "SUP-MASTER ".
005900 01  DS-INVENTORY            PIC X(10)  VALUE "INVENTORY ".
006000
006100 01  DI-ACCOUNT              PIC X(8)  VALUE "ACCOUNT ".
006200 01  DI-DATE-INFO            PIC X(11)  VALUE "DATE-INFO ".
006300 01  DI-STOCK-NO             PIC X(9)  VALUE "STOCK-NO ".
006400 01  DI-PURCH-DATE           PIC X(11)  VALUE "PURCH-DATE ".
006500 01  DI-DELIV-DATE           PIC X(11)  VALUE "DELIV-DATE ".
```

```
006600 01  DI-SUPPLIER            PIC X(9)  VALUE "SUPPLIER ".
006700 01  DI-LASTSHIPDATE        PIC X(13)  VALUE "LASTSHIPDATE ".
006800
```

Using Multiple Modules

When multiple modules are requested, SDCDE prompts for the module name for each data set:

Module name for IMAGE-DATASET *data-set-name* >

Specify the module name for the data set definition. If you enter [[RETURN]], SDCDE does not generate any code for this data set and continues to the next data set in the database. If you specify the ELEM-QUALIFY parameter of the OPTIONS command, SDCDE responds with one of the following prompts, depending on the value you specify:

Value	Prompt
PREFIX	Element prefix >
SUFFIX	Element suffix >
NONE	No prompt

SDCDE attaches the prefix or suffix characters to the elements contained by the data sets. If you enter a [[RETURN]], SDCDE does not attach a prefix or suffix.

When answers to these prompts are complete, SDCDE starts the source generation process for the data set. If you set the ECHO parameter of the OPTIONS command to ON, then all generated code is displayed on the $STDLIST device before continuing with the next data set.

After generating the source for all data sets, SDCDE asks for the module name for the constants. Note that SDCDE issues this prompt only if you set the CONSTANTS parameter of the OPTIONS command to ON:

Module name for *data-base-name* 's constants >

The database constants generated here include the following:

data name name

database passwords

data set names

search items

The database passwords are obtained from the IMAGE-CLASS entity definitions that are related to the database via IMAGE-DATABASE contains IMAGE-CLASS relationship. SDCDE uses the value of the password attribute on the IMAGE-CLASS entity to create the password constants. The constant values generated here use the image-alias attribute value if one exists. If you enter a [[RETURN]] at the prompt, the constants are not generated.

Example

In the following **multiple modules** example the database name is STORE, ELEM-QUALIFY = PREFIX, and CONSTANT = ON.

```
> GENERATE STORE; ENTITY-TYPE=IMAGE-DATABASE.
Define all data sets in one module (Y/N) > N
Module name for IMAGE-DATASET CUSTOMER > CUSTMOD
Element prefix > CUST-
000100
000200 01  CUSTOMER-DATA.
000300     05  CUST-ACCOUNT          PIC S9(9)  COMP.
000400     05  CUST-LAST-NAME        PIC X(16).
000500     05  CUST-FIRST-NAME       PIC X(10).
000600     05  CUST-MID-INITIAL      PIC X(2).
000700     05  CUST-STREET-ADDRESS   PIC X(26).
000800     05  CUST-CITY             PIC X(12).
000900     05  CUST-STATE            PIC X(2).
001000     05  CUST-ZIP              PIC X(6).
001100     05  CUST-CREDIT-RATING    PIC S9(9)  COMP.
001200
001300
Module name for IMAGE-DATASET DATE-MASTER > DATEMOD
Element prefix > DATE-
000100
000200 01  DATE-MASTER-DATA.
000300     05  DATE-DATE-INFO        PIC X(6).
000400
000500
Module name for IMAGE-DATASET PRODUCT > PRODMOD
Element prefix > PRODUCT-
000100
000200 01  PRODUCT-DATA.
000300     05  PRODUCT-STOCK-NO      PIC X(8).
000400     05  PRODUCT-DESCRIPTION   PIC X(20).
000500
000600
Module name for IMAGE-DATASET SALES > SALESMOD
Element prefix > SALES-
```

```
000100
000200 01  SALES-DATA.
000300     05  SALES-ACCOUNT          PIC S9(9)  COMP.
000400     05  SALES-STOCK-NO         PIC X(8).
000500     05  SALES-QUANTITY         PIC S9(4)  COMP.
000600     05  SALES-PRICE            PIC S9(9)  COMP.
000700     05  SALES-TAX              PIC S9(9)  COMP.
000800     05  SALES-TOTAL            PIC S9(9)  COMP.
000900     05  SALES-PURCH-DATE       PIC X(6).
001000     05  SALES-DELIV-DATE       PIC X(6).
001100
001200
Module name for IMAGE-DATASET SUP-MASTER > SUPMOD
Element prefix > SUPPLY-
000100
000200 01  SUP-MASTER-DATA.
000300     05  SUPPLY-SUPPLIER        PIC X(16).
000400     05  SUPPLY-STREET-ADDRESS   PIC X(26).
000500     05  SUPPLY-CITY            PIC X(12).
000600     05  SUPPLY-STATE           PIC X(2).
000700     05  SUPPLY-ZIP             PIC X(6).
000800
000900
Module name for IMAGE-DATASET INVENTORY > INVMOD
Element prefix > INVENT-
000100
000200 01  INVENTORY-DATA.
000300     05  INVENT-STOCK-NO        PIC X(8).
000400     05  INVENT-ONHANDQTY       PIC S9(9)  COMP.
000500     05  INVENT-SUPPLIER        PIC X(16).
000600     05  INVENT-UNIT-COST       PIC S9(7)  COMP-3.
000700     05  INVENT-LASTSHIPDAT     PIC X(6).
000800     05  INVENT-BINNUM          PIC S9(2).
000900
```

```
001000
Module name for STORE'S constants > CONSTMOD
000100
000200 01  DB-STORE              PIC X(8)  VALUE "  STORE ".
000300 01  PWD-STORE             PIC X(10)  VALUE "PASSWORD ".
000400
000500 01  DS-CUSTOMER            PIC X(9)  VALUE "CUSTOMER ".
000600 01  DS-DATE-MASTER          PIC X(12)  VALUE "DATE-MASTER ".
000700 01  DS-PRODUCT            PIC X(8)  VALUE "PRODUCT ".
000800 01  DS-SALES             PIC X(6)  VALUE "SALES ".
000900 01  DS-SUP-MASTER          PIC X(11)  VALUE "SUP-MASTER ".
001000 01  DS-INVENTORY           PIC X(10)  VALUE "INVENTORY ".
001100
001200 01  DI-ACCOUNT            PIC X(8)  VALUE "ACCOUNT ".
001300 01  DI-DATE-INFO          PIC X(11)  VALUE "DATE-INFO ".
001400 01  DI-STOCK-NO           PIC X(9)  VALUE "STOCK-NO ".
001500 01  DI-PURCH-DATE          PIC X(11)  VALUE "PURCH-DATE ".
001600 01  DI-DELIV-DATE         PIC X(11)  VALUE "DELIV-DATE ".
001700 01  DI-SUPPLIER           PIC X(9)  VALUE "SUPPLIER ".
001800 01  DI-LASTSHIPDATE         PIC X(13)  VALUE "LASTSHIPDATE ".
001900
```

Generating IMAGE-DATASET Definitions

If you specify IMAGE-DATASET as the entity type in the GENERATE command, SDCDE generates a COBOL record definition for the data set with the given entity name. SDCDE prompts you for a module name:

Module name for IMAGE-DATASET *data-set-name* >

If you enter a [[RETURN]], SDCDE does not generate any code for this data set and the GENERATE command will be complete. If you enter a module name, SDCDE generates the source into the specified module in the WORKING-STORAGE SECTION.

If you specify the ELEM-QUALIFY parameter of the OPTIONS command, SDCDE responds with one of the following prompts, depending on the value you specify:

Value	Prompt
PREFIX	Element prefix >
SUFFIX	Element suffix >

Value	Prompt
NONE	No prompt

SDCDE attaches the prefix or suffix characters to the elements contained by the data sets. If you enter a [[RETURN]], SDCDE does not a attach prefix or suffix.

When you complete the answers to these prompts, SDCDE starts the source generation process. If you set the SECONDARY-REC parameter of the OPTIONS command to ON, then SDCDE looks for all records related to the data set, regardless of whether the record is primary or secondary. For each such record found, SDCDE generates an 01 level definition with all the elements with the 05 level number. If you set the SECONDARY-REC parameter to OFF, SDCDE uses the data set name to generate the 01 level definition with only the primary record contained in the data set. A complete description of how SDCDE determines the primary record is located in Chapter 5 of this manual.

If you set the CONSTANTS parameter of the OPTIONS command to ON, SDCDE generates the data set name and the search item name as constants. The constant values generated are taken from the image-alias attribute value if one exists. If not, the primary name is used, regardless of the ALIAS parameter value of the OPTIONS command. SDCDE obtains the data set search item name through either the IMAGE-DATASET key ELEMENT relationship or the five-way relationship, depending on whether the data set is a master or detail data set. The five-way relationship consists of IMAGE-DATASET, ELEMENT, ELEMENT, IMAGE-DATASET, and IMAGE-DATABASE chains.

Example

In the following example, the data set name is CUSTOMER, ELEM-QUALIFY = PREFIX, and CONSTANT = ON.

```
> GENERATE CUSTOMER; ENTITY-TYPE=IMAGE-DATASET.

Module name for IMAGE-DATASET CUSTOMER > CUSTMOD

Element prefix > [[RETURN]]

000100
000200 01 CUSTOMER-DATA
000300    05 ACCOUNT              PIC S9(9)  COMP.
000400    05 LAST-NAME             PIC X(16).
000500    05 FIRST-NAME            PIC X(10).
000600    05 MID-INITIAL           PIC X(2).
000700    05 STREET-ADDRESS          PIC X(26).
000800    05 CITY                PIC X(12).
000900    05 STATE                 PIC X(2).
001000    05 ZIP                 PIC X(6).
001200    05 CREDIT-RATING           PIC S9(9)  COMP.
001300
001400 01 DS-CUSTOMER              PIC X(9)  VALUE "CUSTOMER ".
001500 01 DI-ACCOUNT               PIC X(8)  VALUE "ACCOUNT ".
```

001600

Generating IMAGE-DATASET Definitions

If you specify IMAGE-DATASET as the entity type in the GENERATE command, SDCDE generates a COBOL record definition for the data set with the given entity name. SDCDE prompts you for a module name:

Module name for IMAGE-DATASET *data-set-name* >

If you enter a [[RETURN]], SDCDE does not generate any code for this data set and the GENERATE command will be complete. If you enter a module name, SDCDE generates the source into the specified module in the WORKING-STORAGE SECTION.

If you specify the ELEM-QUALIFY parameter of the OPTIONS command, SDCDE responds with one of the following prompts, depending on the value you specify:

Value	Prompt
PREFIX	Element prefix >
SUFFIX	Element suffix >
NONE	No prompt

SDCDE attaches the prefix or suffix characters to the elements contained by the data sets. If you enter a [[RETURN]], SDCDE does not a attach prefix or suffix.

When you complete the answers to these prompts, SDCDE starts the source generation process. If you set the SECONDARY-REC parameter of the OPTIONS command to ON, then SDCDE looks for all records related to the data set, regardless of whether the record is primary or secondary. For each such record found, SDCDE generates an 01 level definition with all the elements with the 05 level number. If you set the SECONDARY-REC parameter to OFF, SDCDE uses the data set name to generate the 01 level definition with only the primary record contained in the data set. A complete description of how SDCDE determines the primary record is located in Chapter 5 of this manual.

If you set the CONSTANTS parameter of the OPTIONS command to ON, SDCDE generates the data set name and the search item name as constants. The constant values generated are taken from the image-alias attribute value if one exists. If not, the primary name is used, regardless of the ALIAS parameter value of the OPTIONS command. SDCDE obtains the data set search item name through either the IMAGE-DATASET key ELEMENT relationship or the five-way relationship, depending on whether the data set is a master or detail data set. The five-way relationship consists of IMAGE-DATASET, ELEMENT, ELEMENT, IMAGE-DATASET, and IMAGE-DATABASE chains.

Example

In the following example, the data set name is CUSTOMER, ELEM-QUALIFY = PREFIX, and CONSTANT = ON.

```
> GENERATE CUSTOMER; ENTITY-TYPE=IMAGE-DATASET.

Module name for IMAGE-DATASET CUSTOMER > CUSTMOD

Element prefix > [[RETURN]]

000100

000200 01 CUSTOMER-DATA
```

```
000300    05 ACCOUNT              PIC S9(9)  COMP.
000400    05 LAST-NAME            PIC X(16).
000500    05 FIRST-NAME           PIC X(10).
000600    05 MID-INITIAL          PIC X(2).
000700    05 STREET-ADDRESS       PIC X(26).
000800    05 CITY                 PIC X(12).
000900    05 STATE                PIC X(2).
001000    05 ZIP                  PIC X(6).
001200    05 CREDIT-RATING        PIC S9(9)  COMP.
001300
001400 01 DS-CUSTOMER             PIC X(9)  VALUE "CUSTOMER ".
001500 01 DI-ACCOUNT              PIC X(8)  VALUE "ACCOUNT ".
001600
```

Generating FORMSFILE Definitions

If you specify FORMSFILE as the entity type in the GENERATE command, SDCDE generates a complete set of COBOL record definitions for the forms contained in the given forms file. To determine whether to generate all forms and constants in one module, or in separate modules, SDCDE issues the following prompt:

```
Define all forms in one module (Y/N) >
```

If you enter a [[RETURN]] or a Y, SDCDE writes all generated code to a single module. If you set the CONSTANTS parameter of the OPTIONS command to ON, the generated code also includes the forms file name, lockword, headform name, and each form name. If you enter an N, each form is written to a separate module, and if the CONSTANTS parameter is set to ON, the forms file name, lockword, and headform name are written to their own module. The SDCDE prompts depend on the choice of single or multiple module options.

Using a Single Module

Once the single module option has been chosen, SDCDE issues a prompt requesting the module name:

```
Module name for FORMSFILE forms-file-name >
```

If the given module name already exists in the current copy library, the copy library's OPEN-MODE parameter determines whether to append all generated code to the existing module or to replace the existing module. If you enter a [[RETURN]] for the module name, SDCDE does not generate any code for the forms file and the GENERATE command will be complete. If you specify the ELEM-QUALIFY parameter of the OPTIONS command, SDCDE responds with one of the following prompts, depending on the value you specify:

Value	Prompt
PREFIX	Element prefix >
SUFFIX	Element suffix >

Value	Prompt
NONE	No prompt

SDCDE attaches the prefix or suffix characters to the elements contained by the forms in the forms file. If you enter a [[RETURN]], SDCDE does not attach a prefix or suffix.

If you set the CONSTANTS parameter of the OPTIONS command to ON, SDCDE issues a prompt for generating a field number table for each form in the forms file. This prompt has the following form:

Define field number tables (N/Y) >

If you enter a [[RETURN]] or an N, SDCDE does not define tables. If you enter a Y, SDCDE defines a field number table following each form.

Each form's constants follow the form's definition. The forms file constants follow the last form. (See "Using Multiple Modules" and "Generating FORM Definitions" for further explanation.)

Note that the constants generated for the forms file name and the form names are taken from the vplus-alias attribute value if one exists, and from the primary name if no vplus-alias attribute value exists.

In all cases SDCDE issues a prompt to determine if the numeric fields of each form should be defined with a PIC 9 or a PIC X. Note that the elements defined as numeric in System Dictionary only appear as ASCII fields in the VPLUS buffer. SDCDE issues the following prompt to determine whether these numeric fields have a PIC 9 or a PIC X description:

Define numeric fields as PIC 9 or PIC X (9/X) >

If you enter a [[RETURN]] or a 9, SDCDE uses a PIC 9 description, otherwise, a PIC X description is used.

When answers to the prompts are complete, SDCDE generates the code in the specified module for the forms file definition. If you set the QUALIFY parameter of the OPTIONS command to ON, SDCDE prefixes all forms with the forms file name. If you set the COMMON-STORE parameter of the OPTIONS command to ON, all forms occupy the same memory space via the REDEFINES clause.

Example

In the following **single module** example, the forms file name is ORDERFF, ELEM-QUALIFY = PREFIX, and CONSTANTS = ON.

```
Define all forms in one module (Y/N) > Y

Module name for FORMSFILE ORDERFF > FORMFILE

Element prefix > [[RETURN]]

Define field number tables (N/Y) > N

Define numeric fields as PIC 9 or PIC X (9/X) > 9

000100
000200  01  CUSTOMER-FORM-DATA.
000300      05  CUST-ACCT          PIC X(8).
000400      05  CUST-NAME          PIC X(30).
000500      05  CUST-ADDRESS       PIC X(60).
000600
```

```
000700  01  FORM-CUSTOMER-FORM       PIC X(14)  VALUE "CUSTOMER-FORM ".
000800
000900  01  ORDER-FORM-DATA.
001000      05  PROD-NO             PIC X(15).
001100      05  QUANTITY            PIC 9(8).
001200      05  UNIT-PRICE          PIC X(12).
001300      05  TOTAL              PIC X(12).
001400
001500  01  FORM-ORDER-FORM         PIC X(11)  VALUE "ORDER-FORM ".
001600
001700  01  FF-ORDERFF              PIC X(8)   VALUE "ORDERFF ".
001800  01  LWD-ORDERFF             PIC X(6)   VALUE "lockff".
001900  01  HF-ORDERFF              PIC X(11)  VALUE "ORDER-FORM ".
002000
```

Using Multiple Modules

When you choose the multiple modules option, SDCDE issues a prompt requesting the module name for the form. This prompt has the following form:

Module name for FORM *form-name* >

If you enter a [[RETURN]] for the module name, SDCDE does not generate any code for the form, and continues to the next form in the forms file.

If you specify the ELEM-QUALIFY parameter of the OPTIONS command, SDCDE responds with one of the following prompts, depending on the value you specify:

Value	Prompt
PREFIX	Element prefix >
SUFFIX	Element suffix >
NONE	No prompt

SDCDE attaches the prefix or suffix characters to the elements contained by the current form. If you enter a [[RETURN]], SDCDE does not attach prefix or suffix.

If you set the CONSTANTS parameter of the OPTIONS command to ON, SDCDE issues a prompt for generating a field number table for the current form. This prompt has the following form:

Define field number tables (N/Y) >

If you enter a [[RETURN]] or an N, SDCDE does not define a table. If you enter a Y, SDCDE defines a field number table for the form. In all cases, SDCDE issues a prompt to determine if the numeric fields of the form should be defined with a PIC 9 or a PIC X. The prompt has the following form:

Define numeric fields as PIC 9 or PIC X (9/X) >

If you enter a [[RETURN]] or a 9, SDCDE uses a PIC 9 description. If you enter an X, SDCDE uses a PIC X description.

When answers to these prompts are complete, SDCDE generates the code in the specified module for the current form. If the current forms' field table and form name constants are to be generated, they appear at the end of the module. If you set the QUALIFY parameter of the OPTIONS command to ON, SDCDE prefixes the form's name with the forms file name. If you set the COMMON-STORE parameter of the OPTIONS command to ON, all forms occupy the same memory space via the REDEFINES clause. If you set the ECHO parameter of the OPTIONS command to ON, all generated code is displayed on the $STDLIST device before continuing with the next form.

The previous steps are repeated for every form in the forms file. When all forms have been defined, and if you set the CONSTANTS parameter to ON, SDCDE issues a prompt requesting the module name for the forms file name, lockword, and head form name constants. The prompt has the following form:

Module name for FORMSFILE *forms-file-name* 's constants >

The forms file constants generated here include the forms file name, its lockword (if any), and the head form's name. The lockword is obtained from the lockword attribute of the FORMSFILE entity. SDCDE uses the head-form attribute of the FORMSFILE contains FORM relationship to determine the head form name. SDCDE uses the vplus-alias attribute value to determine the form name, forms file name, and head form name. Determination of the head form is similar to that of the primary record determination process described in Chapter 5. If you enter a [[RETURN]], SDCDE does not generate any constants.

If a module name already exists in the current copy library, the copy library's OPEN-MODE parameter determines whether to append all generated code to the existing module or to replace the existing module.

Example

In the following **multiple module** example, the forms file name is ORDERFF, ELEM-QUALIFY = SUFFIX, and CONSTANTS = ON.

```
Define all forms in one module (Y/N) > N

Module name for FORM ORDER-FORM-ONE > ORDERFF1

Element suffix > -F1

Define field number tables (N/Y) > Y

Define numeric fields as PIC 9 or PIC X (9/X) > 9

000100
000200  01  CUSTOMER-FORM-DATA.
000300      05  CUST-ACCT-F1          PIC X(8).
000400      05  CUST-NAME-F1          PIC X(30).
000500      05  CUST-ADDRESS-F1       PIC X(60).
000600
000700  01  CUSTOMER-FORM-FIELDS.
000800      05  FIELDNO-CUST-ACCT-F1   PIC S9(4)  COMP  VALUE 1.
000900      05  FIELDNO-CUST-NAME-F1   PIC S9(4)  COMP  VALUE 2.
001000      05  FIELDNO-CUST-ADDRESS-F1   PIC S9(4)  COMP  VALUE 3.
001100
```

```
001200  01  FORM-CUSTOMER-FORM       PIC X(14)  VALUE "CUSTOMER-FORM ".
001300
Module name for FORM ORDER-FORM-TWO > ORDERFF2
Element suffix > -F2
Define field number tables (N/Y) > Y
Define numeric fields as PIC 9 or PIC X (9/X) > 9
000100
000200  01  ORDER-FORM-DATA.
000300      05  PROD-NO-F2          PIC X(15).
000400      05  QUANTITY-F2         PIC 9(8).
000500      05  UNIT-PRICE-F2       PIC X(12).
000600      05  TOTAL-F2            PIC X(12).
000700
000800  01  ORDER-FORM-FIELDS.
000900      05  FIELDNO-PROD-NO-F2    PIC S9(4)  COMP  VALUE 1.
001000      05  FIELDNO-QUANTITY-F2   PIC S9(4)  COMP  VALUE 2.
001100      05  FIELDNO-UNIT-PRICE-F2 PIC S9(4)  COMP  VALUE 3.
001200      05  FIELDNO-TOTAL-F2      PIC S9(4)  COMP  VALUE 4.
001300
001400  01  FORM-ORDER-FORM         PIC X(11)  VALUE "ORDER-FORM ".
001500
Module name for FORMSFILE ORDERFF's constants > ORDERFFC
000100
000200  01  FF-ORDERFF              PIC X(8)  VALUE "ORDERFF ".
000300  01  LWD-ORDERFF             PIC X(6)  VALUE "lockff".
000400  01  HF-ORDERFF              PIC X(11)  VALUE "CUSTOMER-FORM ".
000500
```

Generating FORM Definitions

If you specify FORM as the entity type in the GENERATE command, SDCDE generates a COBOL record definition for the form with the given entity name. SDCDE first prompts you for a module name:

Module name for FORM *form-name* >

If the given module name already exists in the current copy library, the copy library's OPEN-MODE parameter determines whether to append all generated code to the existing module or to replace the existing module. If you enter a [[RETURN]], SDCDE does not generate any code for the record and the GENERATE command will be complete.

If you specify the ELEM-QUALIFY parameter of the OPTIONS command, SDCDE responds with one of the following prompts, depending on the value you specify:

Value	Prompt
PREFIX	Element prefix >
SUFFIX	Element suffix >
NONE	No prompt

SDCDE attaches the prefix or suffix characters to the elements contained by the form. If you enter a [[RETURN]], SDCDE does not attach prefix or suffix.

If you set the CONSTANTS parameter of the OPTIONS command to ON, SDCDE issues a prompt for generating a field number table for the current form. This prompt has the following form:

Define field number tables (N/Y) >

If you enter a [[RETURN]] or an N, SDCDE does not define a table. If you enter a Y, SDCDE defines a field number table for the form. In all cases, SDCDE issues a prompt to determine if the numeric fields of the form should be defined with a PIC 9 or a PIC X. The prompt has the following form:

Define numeric fields as PIC 9 or PIC X (9/X) >

If you enter a [[RETURN]] or a 9, SDCDE uses a PIC 9 description. If you enter an X, SDCDE uses a PIC X description.

When you complete the answers to these prompts, SDCDE generates the code in the specified module for the form definition. If you set the CONSTANTS parameter to ON, descriptions for a form name and field number table are generated. The form name constant value is taken from the vplus-alias attribute if one exists. If a vplus-alias attribute does not exist, the primary name is used. The field numbers for the field number table are obtained from the field-number attribute of the FORM contains ELEMENT relationship.

Example

In the following example, the form name is CUST-ORDER, ELEM-QUALIFY = PREFIX, and CONSTANTS = ON.

```
Module name for FORM CUST-ORDER > ORDFORM

Element prefix > [[RETURN]]

Define field number tables (N/Y) > Y

Define numeric fields as PIC 9 or PIC X (9/X) > 9

000100
000200  01  CUST-ORDER-DATA.
000300      05  CUST-ACCT          PIC X(8).
000400      05  PROD-NO            PIC X(15).
000500      05  QUANTITY           PIC 9(8).
000600      05  UNIT-PRICE         PIC X(12).
000700      05  TOTAL              PIC X(12).
```

```
000800     05  ORDER-DATE          PIC X(8).
000900     05  ORDER-STATUS        PIC X.
001000
001100 01  FORM-CUST-ORDER         PIC X(11)  VALUE "CUST-ORDER ".
001200
001300 01  CUST-ORDER-FIELDS.
001400     05  FIELDNO-CUST-ACCT     PIC S9(4)  COMP  VALUE 1.
001500     05  FIELDNO-PROD-NO       PIC S9(4)  COMP  VALUE 2.
001600     05  FIELDNO-QUANTITY      PIC S9(4)  COMP  VALUE 3.
001700     05  FIELDNO-UNIT-PRICE    PIC S9(4)  COMP  VALUE 4.
001800     05  FIELDNO-TOTAL         PIC S9(4)  COMP  VALUE 5.
001900     05  FIELDNO-ORDER-DATE    PIC S9(4)  COMP  VALUE 6.
002000     05  FIELDNO-ORDER-STATUS   PIC S9(4)  COMP  VALUE 7.
002100
```

Generating FILE Definitions

If you specify FILE as the entity type in GENERATE command, SDCDE generates a COBOL record definition for the file with the given entity name.

If you set the SELECT-FILE parameter of the OPTIONS command to ON, SDCDE issues prompts requesting the module name for the SELECT statement and the FILE SECTION entry; otherwise SDCDE automatically skips the SELECT/FILE SECTION generation.

Module name for FILE *file-name* 's SELECT statement >

You can enter a module name or a [[RETURN]]. If you enter a [[RETURN]], SDCDE does not generate any code for the SELECT statement. In either case, SDCDE issues the following FILE SECTION prompt:

Module name for FILE *file-name* 's FILE SECTION entry >

If you enter a [[RETURN]], SDCDE does not generate any code for the FILE SECTION. If you enter a module name, SDCDE asks you whether to generate the FD or SD:

Define the *file-name* in FILE SECTION as FD or SD (F/S) >

Respond with an F if you wish to generate an FD statement and an S if an SD statement is desired. Note that a "relative" file must be a data file (FD). Therefore, requesting an S for a relative file generates a warning message.

In all cases, SDCDE issues a prompt for the copy library module name to be assigned to the record layout and any constants generated for the file in WORKING-STORAGE SECTION. The prompt has the following form:

Module name for FILE *file-name* 's WORKING-STORAGE >

If you enter a [[RETURN]], the record layout is defined in the FILE SECTION. If you do not enter a FILE SECTION module, a record layout is not defined.

If you specify the ELEM-QUALIFY parameter of the OPTIONS command, and you define a record layout

in either the FILE SECTION or the WORKING-STORAGE SECTION, SDCDE responds with one of the following prompts, depending on the value you specify:

Value	Prompt
PREFIX	Element prefix >
SUFFIX	Element suffix >
NONE	No prompt

SDCDE attaches the prefix or suffix characters to the elements contained by the file. If you enter a [[RETURN]], SDCDE does not attach a prefix or suffix.

When you complete the answers to these prompts, SDCDE generates the code in the specified module or modules. If any module name already exists in the current copy library, the copy library's OPEN-MODE parameter determines whether to append all generated code to the existing module or to replace the existing module.

If you set the CONSTANTS parameter of the OPTIONS command to ON, and the WORKING-STORAGE SECTION is generated, the filename and lockword constants are generated in the WORKING-STORAGE SECTION. If you set the SECONDARY-REC parameter of the OPTIONS command to OFF, only those records that explicitly redefine the primary record, through a RECORD redefines RECORD relationship, have definitions generated using the REDEFINES clause. If you set the SECONDARY-REC parameter to ON, all records explicitly redefining the primary record, as well as all secondary records contained by the file, through a FILE contains RECORD relationship, have definitions generated using the REDEFINES clause. If you set the QUALIFY parameter to ON and the SECONDARY-REC parameter to ON, then SDCDE prefixes all record names by the file name.

The SELECT statement uses the following attributes from the FILE entity (unless otherwise specified) to complete the ASSIGN and ORGANIZATION clauses:

lockword

file-dev-class

recording-mode

cctl-flag (from the FILE uses DEVICE-CLASS relationship)

file-size

file-type

The FD or SD statement uses the following attributes from the FILE entity:

blocking-min

blocking-max

blocking-units

min-record-size

max-record-size

char-type

recording-format

The WORKING-STORAGE SECTION uses the lockword attribute from the FILE entity.

Example 1

In the following example, the file name is ORDERF, SELECT-FILE = ON, and ELEM-QUALIFY = PREFIX.

```
Module name for FILE ORDERF's SELECT statement > ORDERED
Module name for FILE ORDERF's FILE SECTION entry > ORDERFS
Define the ORDERF in FILE SECTION as FD or SD (F/S) > F
Module name for FILE ORDERF's WORKING-STORAGE > ORDERWS
Element prefix > [[RETURN]]
            Module name is ORDERED
000100
000200  SELECT  ORDERF
000300     ASSIGN  "ORDERF/LOCK, DA, A, DISC(CCTL), 48"
000400     ORGANIZATION SEQUENTIAL.
000500
             Module name is ORDERFS
000100
000200  FD  ORDERF              BLOCK 1 TO 5 RECORDS
000300                     RECORDING MODE V
000400                     RECORD 48 CHARACTERS
000500                     CODE-SET ASCII.
000600
000700  01  ORDERF-REC            PIC X(48).
000800
            Module name is ORDERWS
000100
000200  01  ORDERF-DATA.
000300     05  ORD-CUST-ACCT        PIC X(10).
000400     05  ORD-CUST-NAME        PIC X(25).
000500     05  ORD-ORDER-NO         PIC X(12).
000600     05  ORD-ORDER-STATUS     PIC X.
000700
000800  01  FILE-ORDERF           PIC X(8)  VALUE "ORDERF ".
000900  01  LWD-ORDERF            PIC X(8)  VALUE "LOCK".
```

```
001000
```

Example 2

In the following example, the file name is ORDERF, SELECT-FILE = ON, and ELEM-QUALIFY = PREFIX.

```
Module name for FILE ORDERF's SELECT statement > ORDERED
Module name for FILE ORDERF's FILE SECTION entry > ORDERFS
Define the ORDERF in FILE SECTION as FD or SD (F/S) > F
Module name for FILE ORDERF's WORKING-STORAGE > [[RETURN]]
Element prefix > [[RETURN]]
        Module name is ORDERED
000100
000200  SELECT  ORDERFR
000300     ASSIGN  "ORDERF/LOCK, DA, A, DISC(CCTL), 48"
000400     ORGANIZATION SEQUENTIAL.
000500
        Module name is ORDERFS
000100
000200  FD  ORDERF              BLOCK 1 TO 5 RECORDS
000300                      RECORDING MODE V
000400                      RECORD 48 CHARACTERS
000500                      CODE-SET ASCII.
000600
000700
000800  01  ORDERF-DATA.
000900     05  ORD-CUST-ACCT        PIC X(10).
001000     05  ORD-CUST-NAME        PIC X(25).
001100     05  ORD-ORDER-NO         PIC X(12).
001200     05  ORD-ORDER-STATUS     PIC X.
001300
```

Generating KSAMFILE Definitions

If you specify KSAMFILE as entity name given in the GENERATE command, then SDCDE generates a COBOL record definition for the KSAM file with the given entity name.

If you set the SELECT-FILE parameter of the OPTIONS command to ON, SDCDE issues prompts requesting the module name for the SELECT statement and the FILE SECTION entry. These prompts have the following format:

Module name for KSAMFILE *ksam-file-name* 's SELECT statement >

Module name for KSAMFILE *ksam-file-name* 's FILE SECTION entry >

If you enter a [[RETURN]] for the SELECT statement prompt, SDCDE does not generate any code for the SELECT statement. If you enter a [[RETURN]] for the FILE SECTION prompt, SDCDE does not generate any code for the FILE SECTION.

In all cases SDCDE issues a prompt requesting the copy library module name to be assigned to the record layout generated for the KSAM file in the WORKING-STORAGE SECTION:

Module name for KSAMFILE *ksam-file-name* 's WORKING-STORAGE >

If you enter a [[RETURN]], SDCDE defines the record layout in the FILE SECTION. If no FILE SECTION module is created, then SDCDE does not define a record layout. If you specify a non-blank entry to this prompt, SDCDE issues the following prompt:

Include KSAM FILETABLE parameter (N/Y) >

Enter a Y to generate the KSAM FILETABLE parameter for the current KSAM file. Use this table structure when invoking intrinsics provided by KSAM for COBOL.

If you specify the ELEM-QUALIFY parameter of the OPTIONS command, and you define a record layout in either the FILE SECTION or the WORKING-STORAGE SECTION, SDCDE responds with one of the following prompts, depending on the value you specify:

Value	Prompt
PREFIX	Element prefix >
SUFFIX	Element suffix >
NONE	No prompt

SDCDE attaches the prefix or suffix characters to all of the elements contained by the KSAM file. If you enter a [[RETURN]], SDCDE does not attach a prefix or suffix.

When you complete the answers to the prompts, SDCDE generates the code for the KSAM file definition. If any module name already exists in the current copy library, the copy library's OPEN-MODE parameter determines whether to append all generated code to the existing module or to replace the existing module.

If you set the CONSTANTS parameter of the OPTIONS command to ON and the WORKING-STORAGE SECTION is generated, the KSAMFILE name and lockword constants are generated in the WORKING-STORAGE SECTION. The KSAMFILE name constant value uses the standard-alias attribute value if one exists. If the SECONDARY-REC option is OFF, only the primary record and those records that explicitly redefine it through a RECORD redefines RECORD relationship, have definitions generated using the REDEFINES clause. If you set the SECONDARY-REC parameter of the OPTIONS command to ON, all records explicitly redefining the primary record, as well as all secondary records contained by the file, through a FILE contains RECORD relationship, have definitions generated using the REDEFINES clause. If you set the QUALIFY parameter of the OPTIONS command to ON, all record names are prefixed by the KSAMFILE name.

SDCDE supports KSAMFILE handling for COBOL and COBOL II. If you use KSAM support provided by COBOL II, only the SELECT statement and FILE SECTION modules are needed. By entering a [[RETURN]] for the WORKING-STORAGE SECTION module, the record layout is generated in the FILE SECTION. If you use the intrinsics supplied by KSAM for use by COBOL, only the record layout in the WORKING-STORAGE SECTION module is needed. No code should be generated for the SELECT statement or the FILE SECTION. The SELECT statement uses the primary-flag and unique attributes of

the KSAMFILE key ELEMENT relationship to generate the KEY clause. The ASSIGN clause uses the primary record name from the KSAMFILE contains RECORD relationship.

The FD statement uses the following attributes from the KSAMFILE entity:

blocking-min

blocking-max

blocking-units

min-record-size

max-record-size

record-format

Example 1

The following example generates a KSAM definition for COBOL II. In this example, the KSAM file name is ORDERK, SELECT-FILE = ON, CONSTANTS = ON, andELEM-QUALIFY = SUFFIX.

```
Module name for KSAMFILE ORDERK's SELECT Statement > ORDERKED
Module name for KSAMFILE ORDERK's FILE SECTION entry > ORDERKS
Module name for KSAMFILE ORDERK's WORKING-STORAGE > [[RETURN]]
Include KSAM FILETABLE parameter (Y/N) > N
Element Suffix > [[RETURN]]
        Module name is ORDERKED
000100
000200 SELECT   ORDERK
000300     ASSIGN  "ORDERK"
000400     ORGANIZATION INDEXED
000500     RECORD ORDER-NO
000600     ALTERNATE RECORD CUST-ACCT DUPLICATES.
000700
        Module name is ORDERKFS
000100
000200 FD  ORDERK               BLOCK 1 TO 5 RECORDS
000300                     RECORDING MODE V
000400                     RECORD 48 CHARACTERS.
000500
000600
000700 01  ORDERK-DATA.
000800     05  CUST-ACCT          PIC X(10).
```

```
000900     05  CUST-NAME          PIC X(25).
001000     05  ORDER-NO           PIC X(12).
001100     05  ORDER-STATUS       PIC X.
001200
001300
```

Example 2

The following example generates a KSAM definition for COBOL. In this example, the KSAM file name is ORDERK, SELECT-FILE = ON, CONSTANTS = ON, and ELEM-QUALIFY = SUFFIX.

```
Module name for KSAMFILE ORDERK's SELECT Statement > [[RETURN]]
Module name for KSAMFILE ORDERK's FILE SECTION entry > [[RETURN]]
Module name for KSAMFILE ORDERK's WORKING-STORAGE > ORDERKWS
Include KSAM FILETABLE parameter (Y/N) > Y
Element Suffix > [[RETURN]]
        Module name is ORDERKWS
000100
000200 01  ORDERK-DATA.
000300     05  CUST-ACCT          PIC X(10).
000400     05  CUST-NAME          PIC X(25).
000500     05  ORDER-NO           PIC X(12).
000600     05  ORDER-STATUS       PIC X.
000700
000800 01  ORDERK-FILETAB.
000900     05  FILENUMBER         PIC S9(4)  COMP  VALUE 0.
001000     05  FILENAME           PIC X(8)  VALUE "ORDERK  ".
001100     05  I-O-TYPE           PIC S9(4)  COMP  VALUE 0.
001200     05  A-MODE             PIC S9(4)  COMP  VALUE 0.
001300     05  PREV-OP            PIC S9(4)  COMP  VALUE 0.
001400
001500 01  KSAM-ORDERK            PIC X(7)  VALUE "ORDERK ".
001600 01  LWD-ORDERK             PIC X(6)  VALUE "SECRET".
001700
```

Generating RECORD Definitions

If you specify RECORD as the entity type in the GENERATE command, then SDCDE generates a COBOL record definition for the record with the given entity name. SDCDE issues a prompt requesting the copy

library module name to be assigned to all code generated for the record. The prompt has the following form:

Module name for RECORD *record-name* >

If the given module name already exists in the current copy library, the copy library's OPEN-MODE parameter determines whether to append all generated code to the existing module or to replace the existing module. If you enter a [[RETURN]], SDCDE does not generate any code for the record and the GENERATE command will be complete.

If you specify the ELEM-QUALIFY parameter of the OPTIONS command, SDCDE responds with one of the following prompts, depending on the value you specify:

Value	Prompt
PREFIX	Element prefix >
SUFFIX	Element suffix >
NONE	No prompt

SDCDE attaches the prefix or suffix characters to all of the elements contained by the record. If you enter a [[RETURN]], SDCDE does not attach a prefix or suffix.

When answers to the prompts are complete, SDCDE generates the code in the specified module for the record definition. If any records explicitly redefine this record, through the RECORD redefines RECORD relationship, definitions are generated using the REDEFINES clause.

Example

In the following example, the record name is EMPLOYEE-RECORD and ELEM-QUALIFY = PREFIX.

```
Module name for RECORD EMPLOYEE-RECORD > EMPREC
Element prefix > EMP-
000100
000200  01  EMPLOYEE-RECORD-DATA.
000300      05  EMP-NAME-ELEM.
000400          10  EMP-FIRST-NAME     PIC X(15).
000500          10  EMP-LAST-NAME      PIC X(20).
000600      05  EMP-SS-NUMBER          PIC X(10).
000700      05  EMP-POSITION-TITLE     PIC X(50).
000800
```

Generating ELEMENT Definitions

If you specify ELEMENT as the entity type in the GENERATE command, then SDCDE generates a COBOL definition for the element with the given entity name.

SDCDE issues a prompt requesting the copy library module name to be assigned to all code generated for the element. The prompt has the following form:

Module name for ELEMENT *element-name* >

If the given module name already exists in the current copy library, the copy library's OPEN-MODE parameter determines whether to append all generated code to the existing module or to replace the existing module. If you enter a [[RETURN]], SDCDE does not generate any code for this element and the GENERATE command will be complete.

If you specify the ELEM-QUALIFY parameter of the OPTIONS command, SDCDE responds with one of the following prompts, depending on the value you specify:

Value	Prompt
PREFIX	Element prefix >
SUFFIX	Element suffix >
NONE	No prompt

SDCDE attaches the prefix or suffix characters to all of the elements contained by the element. If you enter a [[RETURN]], SDCDE does not attach a prefix or suffix.

When answers to the prompts are complete, SDCDE generates the code in the specified module for the element definition.

Example

In the following example, the element name is DATE and ELEM-QUALIFY = PREFIX.

```
Module name for ELEMENT DATE > DATEDESC
Element prefix > DATE-OF-
000100
000200  01  DATE-ELEM.
000300      05  DATE-OF-YEAR         PIC X(4).
000400      05  DATE-OF-MONTH        PIC X(2).
000500      05  DATE-OF-DAY          PIC X(2).
000600
```

A SDCDE Error Messages

The following is a complete list of SDCDE's errors and warnings. They are listed in order by number, with notes describing the cause and action for each error or warning. The messages are divided into the following categories: The list is divided into the following groups:

Message Number	ErrorType
4500-4512	Informative Messages
4520-4565	Parsing Error Messages
4570-4740	Operational Error Messages
4741-4812	Operational Warning Messages
4813-4884	Fatal Error Messages

NOTE Many error messages in this appendix refer to a definition or file which is shown as a generic name, and will be in italics. An example of this is the name *log-file* in error number 4501 (next page). When the actual message is displayed on your terminal, the generic name *log-file* will be replaced by the actual name of your log file.

SDCDE Informative Messages

4500 MESSAGE **Logging disabled (SDWARN 4500)**

CAUSE SDCDE cannot log the commands the user will enter, because the user decided not to overwrite an old SDLOG.

ACTION None necessary, however, if command logging is important, exit the program, RENAME the old SDLOG, and reenter SDCDE.

CAUSE SDCDE cannot log the commands the user will enter, because the SDLOG file could not be opened.

ACTION None necessary, however, if command logging is important, exit the program and solve the SDLOG opening problem before reentering SDCDE.

4501 MESSAGE **New log file log-file is temporary. Save it for future use (SDWARN 4501)**

CAUSE SDCDE is informing the user of the status of the SDLOG file.

ACTION None necessary. The SDLOG file is created as a temporary file. If the user wishes to save the file, the SAVE SDLOG command should be issued before terminating the current MPE session.

4505 MESSAGE **Both VERSION and STATUS were specified. STATUS is ignored(SDWARN 4505)**

CAUSE SDCDE is informing the user that the VERSION and STATUS parameters may not be used together. When they are, the VERSION parameter will be used to open the dictionary and the STATUS parameter will not be used.

ACTION None necessary. The STATUS parameter will default to the status of the specified version.

4510 MESSAGE **COPYLIB definition already exists in dictionary (SDWARN 4510)**

CAUSE A COPYLIB entity definition with the same name already exists in the dictionary.

ACTION None necessary. A dictionary definition exists for a new copy library. If the old file has been purged and will be replaced by this copy library, there will be no problem, otherwise, notify the Dictionary Administrator that there is a discrepancy between the dictionary and the actual environment.

4511 MESSAGE **KSAMFILE def for copylib key already exists in dictionary (SDWARN 4511)**

CAUSE A KSAMFILE entity definition with the same name as the copy library key file already exists in the dictionary.

ACTION None necessary. A dictionary definition exists for a new KSAMFILE. If the old file has been purged and will be replaced by this key file, there will be no problem, otherwise, notify the Dictionary Administrator that there is a discrepancy between the dictionary and the actual environment.

4512 MESSAGE **COPYLIB contains KSAMFILE rel exists in dictionary (SDWARN 4512)**

CAUSE A COPYLIB contains KSAMFILE relationship with the same entities already exists in the dictionary.

ACTION None necessary. A dictionary definition exists for the relationship between the new copy library and its new key file. If the old files have been purged and are being replaced, there will be no problem, otherwise, notify the Dictionary Administrator that there is a discrepancy between the dictionary and the actual environment.

SDCDE Parse Error Messages

4520 MESSAGE **Unknown SDCDE command (SDERR 4520)**

CAUSE The user has given SDCDE a command that it does not recognize.

ACTION The user should give a command that SDCDE recognizes; these commands can be found by using the HELP command, or in the manual.

4521 MESSAGE **Text from the error to the end of command ignored (SDWARN 4521)**

CAUSE SDCDE ignores the rest of the command from the first error.

ACTION Fix the indicated error and execute the command using REDO, or by reentering the command.

4522 MESSAGE **Command ending period missing (SDERR 4522)**

CAUSE No period was entered to end a multi-line command.

ACTION Add a period to the end of the command using REDO, or by reentering the command.

4523 MESSAGE **Expecting equal sign (SDERR 4523)**

CAUSE An equal character, '=', was left out of the command.

ACTION Add the equal character, '=', to the command using REDO, or by reentering the command.

4524 MESSAGE **Parameter separating semi-colon missing (SDERR 4524)**

CAUSE A semicolon character, ';', is needed to separate the parameter clauses.

ACTION Add the semicolon character, ';', to the command using REDO, or by reentering the command.

4525 MESSAGE **Your response is too long (SDERR 4525)**

CAUSE The user entered a response to a prompt that was too long.

ACTION Use the HELP facility or manual to determine the correct responses for this prompt, and reenter a valid response. The help facility may be accessed by entering a question mark, '?', at any prompt.

4526 MESSAGE **Invalid answer (SDERR 4526)**

CAUSE The user entered an invalid response to a prompt.

ACTION All prompts contain a list of the accepted answers in parentheses. If more help is required, enter a question mark, '?', for help, or check the manual. Reenter a valid response to the current prompt.

4530 MESSAGE **Superfluous parameters found after command ending period (SDERR 4530)**

CAUSE Characters were found after the period which terminated the command.

ACTION Omit all characters after the period in the command using REDO, or by reentering the command.

4531 MESSAGE **Found more arguments than command requires (SDERR 4531)**

CAUSE The command contains more arguments than the syntax allows.

ACTION Omit all extraneous parameters from the command using REDO, or by reentering the command. A description of the valid arguments for each command may be found in the HELP facility or the manual.

4532 MESSAGE **Expecting 'Y' or 'N' answer (SDERR 4532)**

CAUSE The user entered a response other than 'Y' or 'N' to the prompt.

ACTION Enter 'Y' or 'N'.

4533 MESSAGE **Mismatched quotes (SDERR 4533)**

CAUSE The quoted password contains mismatching quotes.

ACTION Reenter the password by using the REDO command, or by retyping the whole command. Make sure that the beginning quote has a matching quote within the same command line. Quotes can be included in the password by specifying two consecutive quotes.

4534 MESSAGE **Unknown DEFINE parameter (SDERR 4534)**

CAUSE The parameter being specified is not recognizable by SDCDE.

ACTION Determine the valid DEFINE command parameters using the HELP DEFINE command or the manual. Fix the parameter using REDO, or by reentering the command.

4535 MESSAGE **Expecting dictionary name (SDERR 4535)**

CAUSE The dictionary name was not supplied in the DICTIONARY parameter.

ACTION Add the dictionary name to the DICTIONARY parameter using REDO, or by reentering the command.

4536 MESSAGE **Expecting scope name (SDERR 4536)**

CAUSE The scope name was not supplied in the SCOPE parameter.

ACTION Add the scope name to the SCOPE parameter using REDO, or by reentering the command.

4541 MESSAGE **Expecting version status value (SDERR 4541)**

CAUSE The version status is not supplied in the STATUS parameter.

ACTION Add the version status to the STATUS parameter using REDO, or by reentering the command.

4542 MESSAGE **Illegal dictionary open mode (SDERR 4542)**

CAUSE The open mode supplied in the OPEN-MODE parameter is not valid.

ACTION Add a valid open mode to the OPEN-MODE parameter using REDO, or by reentering the command. The valid open modes are: READ-ONLY, SHARED-UPDATE, and EXCLUSIVE-UPDATE.

4543 MESSAGE **Illegal dictionary name mode (SDERR 4543)**

CAUSE The name mode supplied in the NAME-MODE parameter is not valid.

ACTION Add a valid name mode to the NAME-MODE parameter using REDO, or by reentering the command. The valid name modes are: EXTERNAL and INTERNAL.

4544 MESSAGE **Unknown COPYLIB parameter (SDERR 4544)**

CAUSE The parameter being specified is not recognized as a COPYLIB command parameter.

ACTION A list of valid COPYLIB command parameters can be found by using the HELP COPYLIB command, or in the manual. Fix the parameter using REDO, or by reentering the command.

4545 MESSAGE **Expecting copy library name (SDERR 4545)**

CAUSE No copylib name was supplied to the NAME parameter.

ACTION Add the copy library name to the NAME parameter using REDO, or by reentering the command.

4550 MESSAGE **Expecting ON/OFF switch value for the UPDATE-DICT parm (SDERR 4550)**

CAUSE The UPDATE-DICT parameter accepts a value of ON or OFF only.

ACTION Assign ON or OFF to the parameter using REDO, or by reentering the command.

4551 MESSAGE **Illegal copy library open mode (SDERR 4551)**

CAUSE The open mode supplied in the OPEN-MODE parameter is not valid.

ACTION Add a valid open mode to the OPEN-MODE parameter using REDO, or by reentering the command. The valid open mode values are: APPEND or REPLACE.

4552 MESSAGE **Unknown OPTIONS parameter (SDERR 4552)**

CAUSE The OPTIONS command contains an unrecognizable parameter.

ACTION A list of all parameters for the OPTIONS command can be found using the HELP OPTIONS command, or in the manual.

4553 MESSAGE **Illegal ALIAS value (SDERR 4553)**

CAUSE The value supplied in the ALIAS parameter is not valid.

ACTION Add a valid value to the ALIAS parameter using REDO, or by reentering the command. The valid values are: COBOL, IMAGE, STANDARD, VPLUS, or NONE.

4554 MESSAGE **Expecting ON/OFF switch (SDERR 4554)**

CAUSE The value supplied for the OPTIONS command parameter is invalid.

ACTION The value should be changed to ON or OFF using REDO, or by reentering the command.

4555 MESSAGE **Illegal ELEM-QUALIF value (SDERR 4555)**

CAUSE The value supplied in the ELEM-QUALIFY parameter is not valid.

ACTION Add a valid value to the ELEM-QUALIFY parameter using REDO, or by reentering the command. The valid values are: PREFIX, SUFFIX, or NONE.

4560 MESSAGE **LIST command syntax error (SDERR 4560)**

CAUSE The syntax of the LIST command is invalid.

ACTION See the correct syntax using the HELP LIST command, or the manual. Update the command using REDO, or by reentering the command.

4561 MESSAGE **Entity name missing (SDERR 4561)**

CAUSE The entity name is required in the GENERATE command.

ACTION Add a valid entity name as the first parameter of the GENERATE command using REDO, or by reentering the command.

4562 MESSAGE **ENTITY-TYPE clause missing (SDERR 4562)**

CAUSE The ENTITY-TYPE parameter is required in the GENERATE command.

ACTION Add the ENTITY-TYPE parameter to the GENERATE command using REDO, or by reentering the command.

4563 MESSAGE **Entity type name missing (SDERR 4563)**

CAUSE The entity type name is missing from the ENTITY-TYPE parameter.

ACTION Add the entity type name to the ENTITY-TYPE parameter using REDO, or by reentering the command.

4564 MESSAGE **GENERATE command expecting an ending period (SDERR 4564)**

CAUSE No period was entered to end the GENERATE command.

ACTION Add a period to the end of the command using REDO, or by reentering the command.

4565 MESSAGE **Unknown STANDARD-PARMS parameter (SDERR 4565)**

CAUSE The value of the STANDARD-PARMS command parameter is not valid.

ACTION Change the parameter to a valid value using REDO, or by reentering the command. The valid values are: IMAGE, KSAM, or VPLUS.

SDCDE Operational Error Messages

4570 MESSAGE **Error while opening an old SDLOG (SDERR 4570)**

CAUSE An MPE file system error was detected when attempting to open the existing SDLOG file.

ACTION The commands you enter will not be logged, as the logging is disabled. If command logging is required, exit the program immediately, take care of the file system problem, and reenter SDCDE. Notify the System Manager if a serious error is suspected.

4571 MESSAGE **Error while opening a new SDLOG (SDERR 4571)**

CAUSE A MPE file system error was detected when attempting to open a new SDLOG file.

ACTION The command you enter will not be logged, as the logging is disabled. If command logging is required, exit the program immediately, take care of the file system problem, and reenter SDCDE. Notify the System Manager if a serious error is suspected.

4572 MESSAGE **Could not open the System Dictionary (SDERR 4572)**

CAUSE The specified System Dictionary could not be opened.

ACTION The System Dictionary may be opened exclusively by another user, otherwise, if the problem cannot be solved, notify the Dictionary Administrator.

4573 MESSAGE **Could not close the System Dictionary (SDERR 4573)**

CAUSE The currently opened System Dictionary could not be closed.

ACTION Notify the Dictionary Administrator.

4574 MESSAGE **Could not open the Dictionary because of invalid scope password(SDERR 4574)**

CAUSE The specified password is not valid for the given scope.

ACTION Reenter the correct scope password.

--

4575 MESSAGE **Could not create a new copy library (SDERR 4575)**

CAUSE An MPE file system error occurred while creating a new copy library.

ACTION If the file system error cannot be solved, notify the System Manager.

4576 MESSAGE **Could not close the copy library (SDERR 4576)**

CAUSE An MPE file system error occurred while closing the currently opened copy library.

ACTION If the file system error cannot be solved, notify the System Manager.

4580 MESSAGE **Key file name for a new copy library empty (SDERR 4580)**

CAUSE No key file name was supplied in the KEY-FILE parameter. This is required for a new copy library.

ACTION Add a key file name to the KEY-FILE parameter of the COPYLIB command using REDO, or by reentering the command.

4581 MESSAGE **Copylib definition cannot be added because no dictionary opened(SDERR 4581)**

CAUSE The dictionary has not been opened, therefore, the definition of a new copy library cannot be added to the dictionary through the COPYLIB command.

ACTION The dictionary must be successfully opened using the DEFINE command before attempting to update the dictionary with new copy library definitions from the COPYLIB command.

4582 MESSAGE **Copylib definition cannot be added because of invalid open mode(SDERR 4582)**

CAUSE The dictionary was opened in READ-ONLY mode, therefore, the definition of a new copy library cannot be added to the dictionary through the COPYLIB command.

ACTION The dictionary must be opened in SHARED-UPDATE or EXCLUSIVE-UPDATE mode before it can be updated with new copy library definitions from the COPYLIB command.

4583 MESSAGE **Error while creating copylib definition in dictionary (SDERR4583)**

CAUSE A System Dictionary error occurred while creating an entity definition for the copy library definition.

ACTION Notify the Dictionary Administrator.

4584 MESSAGE **Error while creating copylib key file def in dictionary(SDERR4584)**

CAUSE A System Dictionary error occurred while creating an entity definition for the copy library key file definition.

ACTION Notify the Dictionary Administrator.

4585 MESSAGE **Error while creating COPYLIB contains KSAMFILE rel (SDERR 4585)**

CAUSE A System Dictionary error occurred while creating a relationship definition for the copy library.

ACTION Notify the Dictionary Administrator.

4586 MESSAGE **File specified is not a valid COPYLIB (SDERR 4586)**

CAUSE The file that was specified is not a valid copylib.

ACTION Using KSAMUTIL.PUB.SYS, make sure that the file is a KSAMFILE with the copylib structure. It should have only one byte key of length 14 and starting at 73. The record number should also start from 0 (as opposed to 1), and no duplicate key should be allowed.

4590 MESSAGE **Copylib already accessed by another process (SDERR 4590)**

CAUSE The specified copylib could not be opened, because it is currently being accessed by another process, either exclusively or non-exclusively.

ACTION If the definitions must be generated into the specified copy library, the user must wait until it is not being accessed. If no particular copy library is required for the definitions, open another copy library.

4591 MESSAGE **Error occurred while listing entities of entity type entity-type(SDERR 4591)**

CAUSE A System Dictionary error occurred while retrieving entities for the LIST command.

ACTION Notify the Dictionary Administrator.

4592 MESSAGE **LIST command fails because no dictionary opened (SDERR 4592)**

CAUSE The dictionary must be opened when attempting to list all of the entities of a specific entity type.

ACTION The dictionary must be successfully opened using the DEFINE command before the entities of an entity type may be listed using the LIST command.

4593 MESSAGE **LIST MODULE fails because no copylib opened (SDERR 4593)**

CAUSE The copy library must be opened when attempting to list all of the modules contained in it.

ACTION The copy library must be successfully opened using the COPYLIB command before a listing of the modules within it may be listed using the LIST command.

4594 MESSAGE **Error while running COBEDIT.PUB.SYS (SDERR 4594)**

CAUSE A MPE error occurred while attempting to run the COBEDIT program in the PUB group of the SYS account.

ACTION Verify that COBEDIT.PUB.SYS exists, and that the current user has access to it. If the file exists, and is accessible, notify the System Manager.

4595 MESSAGE **Error while generating standard parameters (SDERR 4595)**

CAUSE Unable to retrieve the standard parameter definition from the message catalog.

ACTION Notify the Dictionary Administrator.

4596 MESSAGE **Entity type invalid or not allowed with LIST command (SDERR4596)**

CAUSE The value given for the entity type parameter is a valid dictionary entity type, but is not supported by this command.

ACTION Reenter the command using a supported entity type or keyword. Valid entity types are: COPYLIB, IMAGE-DATABASE, IMAGE-DATASET, FORMSFILE, FORM, FILE, KSAMFILE, RECORD, and ELEMENT.

CAUSE The value given for the entity type parameter is not a valid dictionary entity type nor a valid keyword.

ACTION Reenter the command using a supported entity type or keyword. Valid entity types are: COPYLIB, IMAGE-DATABASE, IMAGE-DATASET, FORMSFILE, FORM, FILE, KSAMFILE, RECORD, and ELEMENT.

4600 MESSAGE **STANDARD-PARMS command fails because no copylib opened(SDERR4600)**

CAUSE The copy library must be opened in order to write any generated definitions to it.

ACTION The copy library must be successfully opened using the COPYLIB command before the STANDARD-PARMS command may be executed.

4601 MESSAGE **GENERATE command fails because no dictionary opened (SDERR 4601)**

CAUSE The dictionary must be opened in order to generate any definitions from it.

ACTION The dictionary must be successfully opened using the DEFINE command before the GENERATE command may be executed.

4602 MESSAGE **GENERATE command fails because no copy library opened (SDERR4602)**

CAUSE The copy library must be opened in order to write any generated definitions to it.

ACTION The copy library must be successfully opened using the COPYLIB command before the GENERATE command may be executed.

4603 MESSAGE **Entity-type is a non-existent entity type (SDERR 4603)**

CAUSE The entity type name supplied in the ENTITY-TYPE parameter is not valid.

ACTION Change the entity type name in the ENTITY-TYPE parameter to a valid value using

REDO, or by reentering the command. The valid values are: IMAGE-DATABASE, IMAGE-DATASET, FORMSFILE, FORM, FILE, KSAMFILE, RECORD, and ELEMENT.

4604 MESSAGE **Entity of the specified type does not exist (SDERR 4604)**

CAUSE The given entity name in the GENERATE command does not exist for the entity type given in the ENTITY-TYPE parameter.

ACTION The entity name may be misspelled, it may not exist, or the entity type may not be the correct type. Use LIST to get a list of all entities of the given entity type. Use REDO, or reenter the command using the correct parameters.

4605 MESSAGE **SDCDE does not generate COBOL source for entity-type entity type(SDERR 4605)**

CAUSE The entity type name supplied in the ENTITY-TYPE parameter is not one of the entity types supported by SDCDE.

ACTION The entity type name in the ENTITY-TYPE parameter must be changed to a value that is supported by SDCDE. The supported entity type names are: IMAGE-DATABASE, IMAGE-DATASET, FORMSFILE, FORM, FILE, KSAMFILE, RECORD, and ELEMENT.

4606 MESSAGE **Entity type invalid or not allowed with GENERATE command (SDERR4606)**

CAUSE The value given for the entity type parameter is a valid dictionary entity type, but is not supported by this command.

ACTION Reenter the command using a supported entity type. Valid entity types are: IMAGE-DATABASE, IMAGE-DATASET, FORMSFILE, FORM, FILE, KSAMFILE, RECORD, and ELEMENT.

CAUSE The value given for the entity type parameter is not a valid dictionary entity type.

ACTION Reenter the command using a supported entity type or keyword. Valid entity types are: IMAGE-DATABASE, IMAGE-DATASET, FORMSFILE, FORM, FILE, KSAMFILE, RECORD, and ELEMENT.

4610 MESSAGE **Module name specified is too long (SDERR 4610)**

CAUSE The module name entered at the prompt must not exceed 8 characters.

ACTION Reenter a module name that does not exceed 8 characters.

4611 MESSAGE **Module name should consist of alphanumeric characters and '-'(SDERR 4611)**

CAUSE An illegal character was used in the module name. Only alphanumeric characters and '-' are allowed in a module name.

ACTION Reenter a module name that has no illegal characters.

4612 MESSAGE **Module name ca**nnot end with a '-' (SDERR 4612)

CAUSE The module name may contain a '-', but not end with one.

ACTION Reenter a module name that has no '-' as the last character.

4613 MESSAGE **Element qual should consist of alphanumeric chars and '-'(SDERR4613)**

CAUSE An illegal character was used in the element qualifier. Only alphanumeric characters and '-' are allowed.

ACTION Reenter a element qualifier that has no illegal characters.

4614 MESSAGE **Error retrieving description attribute (SDERR 4614)**

CAUSE A System Dictionary error occurred while retrieving the description variable attribute.

ACTION Notify the Dictionary Administrator.

4615 MESSAGE **Unable to retrieve IMAGE-DATABASE contains IMAGE-DATASET fordata-base (SDERR 4615)**

CAUSE A System Dictionary error occurred while retrieving the IMAGE-DATABASE contains IMAGE-DATASET relationship.

ACTION Notify the Dictionary Administrator.

4620 MESSAGE **Unable to retrieve comment related IMAGE-DATABASE attr's fordata-base (SDERR 4620)**

CAUSE A System Dictionary error occurred while retrieving attributes required to comment the given IMAGE-DATABASE entity.

ACTION Notify the Dictionary Administrator.

4621 MESSAGE **Unable to retrieve IMAGE-CLASS attr's for image-class (SDERR4621)**

CAUSE A System Dictionary error occurred while retrieving attributes from the given IMAGE-CLASS entity.

ACTION Notify the Dictionary Administrator.

4623 MESSAGE **Unable to retrieve IMAGE-DATABASE attr's for data-base (SDERR4623)**

CAUSE A System Dictionary error occurred while retrieving attributes for the given database.

ACTION Notify the Dictionary Administrator.

4625 MESSAGE **Unable to find classes contained in IMAGE-DATABASE data-base(SDERR 4625)**

CAUSE A System Dictionary error occurred while finding any IMAGE-DATABASE contains IMAGE-CLASS relationships for the given IMAGE-DATABASE entity.

ACTION Notify the Dictionary Administrator.

4630 MESSAGE **Unable to retrieve comment related IMAGE-DATASET attr's fordata-set(SDERR 4630)**

CAUSE A System Dictionary error occurred while retrieving attributes required to comment the given IMAGE-DATASET entity.

ACTION Notify the Dictionary Administrator.

4631 MESSAGE **Unable to retrieve ELEMENT attr's for search item search-item(SDERR 4631)**

CAUSE A System Dictionary error occurred while retrieving attributes from the given ELEMENT entity.

ACTION Notify the Dictionary Administrator.

4633 MESSAGE **Unable to retrieve IMAGE-DATASET attr's for data-set (SDERR4633)**

CAUSE A System Dictionary error occurred while retrieving attributes from the given IMAGE-DATASET entity.

ACTION Notify the Dictionary Administrator.

4635 MESSAGE **Unable to find RECORDs contained in IMAGE-DATASET data-set(SDERR 4635)**

CAUSE A System Dictionary error occurred while finding any IMAGE-DATASET contains RECORD relationships for the given IMAGE-DATASET.

ACTION Notify the Dictionary Administrator.

4640 MESSAGE **Unable to find data-set's search items (SDERR 4640)**

CAUSE A System Dictionary error occurred while finding any IMAGE-DATASET, ELEMENT, ELEMENT, IMAGE-DATASET, IMAGE-DATABASE chains relationships or IMAGE-DATASET key ELEMENT relationships for the given IMAGE-DATASET entity, depending on whether the data set is a detail set or not.

ACTION Notify the Dictionary Administrator.

4641 MESSAGE **Unable to retrieve FORMSFILE contains FORM for forms-file (SDERR4641)**

CAUSE An Image error was encountered while retrieving the FORMSFILE contains FORM relationship from the dictionary.

ACTION Notify the Dictionary Administrator.

4642 MESSAGE **Unable to retrieve comment related FORMSFILE attr's forforms-file(SDERR 4642)**

CAUSE A System Dictionary error occurred while retrieving attributes required to comment the given FORMSFILE entity.

ACTION Notify the Dictionary Administrator.

4643 MESSAGE **Unable to retrieve FORMSFILE attr's for forms-file (SDERR 4643)**

CAUSE A System Dictionary error occurred while retrieving attributes for the given FORMS-FILE entity.

ACTION Notify the Dictionary Administrator.

4645 MESSAGE **Unable to retrieve FORM contains ELEMENT for form (SDERR 4645)**

CAUSE A System Dictionary error occurred while retrieving the FORM contains ELEMENT relationship for the given FORM.

ACTION Notify the Dictionary Administrator.

4650 MESSAGE **Unable to retrieve comment related FORM attr's for form(SDERR4650)**

CAUSE A System dictionary error occurred while retrieving attributes required to comment the given FORM entity.

ACTION Notify the Dictionary Administrator.

4651 MESSAGE **Unable to retrieve FORM attr's for form (SDERR 4651)**

CAUSE A System Dictionary error occurred while retrieving attributes for the given FORM entity.

ACTION Notify the Dictionary Administrator.

4653 MESSAGE **Unable to retrieve ELEMENT attr's for element to form fieldtable (SDERR 4653)**

CAUSE A System Dictionary error occurred while retrieving attributes for the ELEMENT contained in the FORM to generate the field number table.

ACTION Notify the Dictionary Administrator.

4655 MESSAGE **Unable to retrieve comment related FILE attr's for file (SDERR4655)**

CAUSE A System Dictionary error occurred while retrieving attributes required to comment the given FILE entity.

ACTION Notify the Dictionary Administrator.

4660 MESSAGE **Unable to retrieve FILE attr's for file (SDERR 4660)**

CAUSE A System Dictionary error occurred while retrieving attributes for the given FILE entity.

ACTION Notify the Dictionary Administrator.

4662 MESSAGE **Unable to retrieve DEVICE-CLASS attr's for device-class (SDERR4662)**

CAUSE A System Dictionary error occurred while retrieving the attributes associated with the FILE uses DEVICE-CLASS relationship for the given FILE entity.

ACTION Notify the Dictionary Administrator.

4663 MESSAGE **Unable to find RECORD's related to the FILE file (SDERR 4663)**

CAUSE A System Dictionary error occurred while retrieving the FILE contains RECORD relationship for the given FILE entity.

ACTION Notify the Dictionary Administrator.

4664 MESSAGE **Unable to retrieve comment related KSAMFILE attr's for ksamfile (SDERR 4664)**

CAUSE A System Dictionary error occurred while retrieving attributes required to comment the given KSAMFILE entity.

ACTION Notify the Dictionary Administrator.

4665 MESSAGE **Unable to retrieve KSAMFILE attr's for ksamfile (SDERR 4665)**

CAUSE A System Dictionary error occurred while retrieving attributes for the given KSAMFILE entity.

ACTION Notify the Dictionary Administrator.

4671 MESSAGE **Unable to retrieve KSAMFILE key attr's for key-item (SDERR 4671)**

CAUSE A System Dictionary error occurred while retrieving attributes for the given KSAMFILE file's keys.

ACTION Notify the Dictionary Administrator.

4672 MESSAGE **Unable to find keys related to the KSAMFILE ksamfile (SDERR4672)**

CAUSE A System Dictionary error occurred while finding any KSAMFILE key ELEMENT relationships for the given KSAMFILE entity.

ACTION Notify the Dictionary Administrator.

4673 MESSAGE **Unable to find RECORD's related to the KSAMFILE ksamfile (SDERR4673)**

CAUSE A System Dictionary error occurred while finding any KSAMFILE contains RECORD

relationships for the given KSAMFILE entity.

ACTION Notify the Dictionary Administrator.

4674 MESSAGE **Unable to produce KSAM file table (SDERR 4674)**

CAUSE Unable to retrieve the KSAM file table definition from the message catalog.

ACTION Notify the Dictionary Administrator.

4675 MESSAGE **Unable to retrieve RECORD contains ELEMENT for record (SDERR4675)**

CAUSE A System Dictionary error occurred while retrieving any RECORD contains ELEMENT relationships for the given RECORD entity.

ACTION Notify the Dictionary Administrator.

4680 MESSAGE **Unable to retrieve RECORD redefines RECORD for second-record (SDERR 4680)**

CAUSE A System Dictionary error occurred while retrieving any RECORD redefines RECORD relationships for the given RECORD entity.

ACTION Notify the Dictionary Administrator.

4681 MESSAGE **Unable to retrieve comment related attr's for RECORD record (SDERR 4681)**

CAUSE A System Dictionary error occurred while retrieving attributes for required to comment given RECORD entity.

ACTION Notify the Dictionary Administrator.

4682 MESSAGE **Unable to retrieve RECORD attr's for record (SDERR 4682)**

CAUSE A System Dictionary error occurred while retrieving attributes for the given RECORD entity.

ACTION Notify the Dictionary Administrator.

4684 MESSAGE **Unable to retrieve ELEMENT contains ELEMENT for first-element (SDERR 4684)**

CAUSE A System Dictionary error occurred while retrieving any ELEMENT contains ELEMENT relationships for the given ELEMENT entity.

ACTION Notify the Dictionary Administrator.

4685 MESSAGE **Unable to retrieve ELEMENT redefines ELEMENT for second-element (SDERR 4685)**

CAUSE A System Dictionary error occurred while retrieving any ELEMENT redefines ELEMENT relationships for the given ELEMENT entity.

ACTION Notify the Dictionary Administrator.

4689 MESSAGE **Unable to retrieve ELEMENT contains CONDITION-NAME for element (SDERR 4689)**

CAUSE 1. A System Dictionary error occurred while retrieving an ELEMENT contains CONDITION-NAME relationship for the given ELEMENT entity. 2. SDCDE version may not be compatible with the current version of the System Dictionary core set.

ACTION 1. Notify the Dictionary Administrator. 2. Notify the Dictionary Administrator.

4690 MESSAGE **Unable to retrieve ELEMENT references ELEMENT for first-element (SDERR 4690)**

CAUSE A System Dictionary error occurred while retrieving any ELEMENT references ELEMENT relationships for the given ELEMENT entity.

ACTION Notify the Dictionary Administrator.

4691 MESSAGE **Unable to retrieve comment attr's for ELEMENT element (SDERR4691)**

CAUSE A System Dictionary error occurred while retrieving attributes required to comment the given ELEMENT entity.

ACTION Notify the Dictionary Administrator.

4692 MESSAGE **Maximum recursive level. Generation terminated at entity (SDERR4692)**

CAUSE The ELEMENT contains ELEMENT causes a nested definition. Only 30 levels of contains are allowed before generation automatically terminates.

ACTION Definitions that contain multiple levels of nesting can be inefficient as well as indecipherable. The ELEMENT contains ELEMENT relationships should not be nested too deeply in the dictionary.

4693 MESSAGE **Recursive defn in System Dictionary for entity (SDERR 4693)**

CAUSE The definition being generated has found that the given entity is defined by itself, either directly or indirectly.

ACTION The given entity must not have any directly recursive relationships: ELEM1 contains ELEM1, ELEM1 redefines ELEM1, or ELEM1 references ELEM1; or any indirectly recursive relationships: ELEM1 contains ELEM2, ELEM2 redefines ELEM1.

4694 MESSAGE **Recursive defn in COBOL source for cobol-name (SDERR 4694)**

CAUSE The definition being generated already contains the given COBOL name at the same level number, or as a parent.

ACTION An alias name may be used, or the entity name may be altered in the dictionary. If the qualification is causing the recursive definition, then omit or shorten the qualification.

4695 MESSAGE **Recursive defn in System Dictionary and COBOL source for entity (SDERR 4695)**

CAUSE The definition being generated has found that the given entity is defined by itself, either directly or indirectly. The entity is also using the same COBOL name as a previously defined COBOL definition.

ACTION See both SDERR 4693 and SDERR 4694.

4700 MESSAGE **Unable to retrieve attributes for ELEMENT element (SDERR 4700)**

CAUSE A System Dictionary error occurred while retrieving attributes for the given ELEMENT entity.

ACTION Notify the Dictionary Administrator.

4701 MESSAGE **Unexpected or missing element type (SDERR 4701)**

CAUSE The element type attribute retrieved from the dictionary is blank or does not contain a valid COBOL element type.

ACTION The dictionary element type attribute must be updated to one of the following valid

values: B, E, E+, I, I+, J, J+, K, P, P+, R, R+, S, X, U, Z, Z+, 9, 9+, *.

4702 MESSAGE **Illegal attribute, decimal is negative (SDERR 4702)**

CAUSE The decimal attribute retrieved from the dictionary has an invalid value. It must be zero or a positive integer.

ACTION The dictionary decimal attribute must be updated to zero or a positive integer.

4703 MESSAGE **Illegal attribute, display length is zero or negative (SDERR4703)**

CAUSE The display length attribute retrieved from the dictionary has an invalid value. It must be a positive integer.

ACTION The dictionary display length attribute must be updated to a positive integer.

4704 MESSAGE **Illegal attribute, byte length is zero or negative (SDERR 4704)**

CAUSE The byte length attribute retrieved from the dictionary has an invalid value. It must be a positive integer.

ACTION The dictionary byte length attribute must be updated to a positive integer.

4705 MESSAGE Ca**lculated element length is zero or negative (SDERR 4705)**

CAUSE The length of the picture is invalid when calculated from the byte-length and display-length attributes.

ACTION Adjust the byte-length and/or display-length attributes in the dictionary, so that the picture will be calculated as a positive integer.

4710 MESSAGE **Illegal attribute, decimallength exceeds element length (SDERR4710)**

CAUSE The value of the decimal attribute is greater than the calculated length of the element (using the byte length and/or display length attributes).

ACTION The decimal, byte length, and display length attributes must be reviewed and updated in the dictionary.

4711 MESSAGE **Illegal attribute, byte offset is negative (SDERR 4711)**

CAUSE The byte offset attribute retrieved from the dictionary has an invalid value. It must be zero or a positive integer.

ACTION The dictionary byte offset attribute must be updated to zero or a positive integer.

4712 MESSAGE **Illegal attribute for element, count not allowed with redefines (SDERR 4712)**

CAUSE The count attribute contains a positive integer greater than one, indicating use of the occurs clause. However, the occurs clause cannot be used on an element that redefines another element, or is contained in an element or record that redefines another element or record.

ACTION The dictionary count attribute must be updated to one, or the element must not be a part of a redefining element or record.

4713 MESSAGE **Illegal attribute, count is zero or negative (SDERR 4713)**

CAUSE The count attribute retrieved from the dictionary has an invalid value. It must be a positive integer.

ACTION The dictionary count attribute must be updated to a positive integer.

4714 MESSAGE **Illegal occurs clause, table exceeds 3 dimensions (SDERR 4714)**

CAUSE The count attribute has a value greater than one, indicating that an occurs clause should be used. Since this element is already contained in an element which is the third level of an array, COBOL does not allow a forth level.

ACTION The dictionary count attribute must be updated to one, or the element removed from the array.

4715 MESSAGE **No command to REDO (SDERR 4715)**

CAUSE No command has been issued in SDCDE prior to entering the REDO command, i.e., the REDO is issued as the first command to SDCDE.

ACTION The REDO command is not valid unless a prior command has been entered. Prior commands do not include empty lines, control Y, or CR.

4720 MESSAGE **REDO is not allowed in non-interactive mode (SDERR 4720)**

CAUSE SDCDE does not allow the REDO command when the input comes from SDIN, or if SDCDE is being run in a job.

ACTION Remove the REDO command from the input file.

4721 MESSAGE **Illegal use of REDO (SDERR 4721)**

CAUSE The REDO command cannot be issued from within the REDO command.

ACTION Do not use REDO to create a REDO command.

4722 MESSAGE **Command line out of range. Ignored (SDERR 4722)**

CAUSE The REDO sub-command attempted to move to a line beyond the beginning or end of the command being edited.

ACTION List the command being edited, and determine which line you wish to move to. Reenter the correct number of lines to move.

4723 MESSAGE **Illegal command after delete. Ignored (SDERR 4723)**

CAUSE The delete sub-command character(s) of the REDO command must be followed by blanks or by the insert sub-command character.

ACTION Modify the command so that the delete sub-command characters(s) are followed only by blanks or the insert sub-command character, 'I'.

4724 MESSAGE **String to insert too long to fit (SDERR 4724)**

CAUSE The string which follows the insert sub-command will cause the current line to exceed 80 characters.

ACTION Reenter the insert sub-command using less characters, or break the edited line into two lines and then reenter the insert sub-command.

4725 MESSAGE **String to append too long to fit (SDERR 4725)**

CAUSE The string which follows the append sub-command will cause the current line to exceed 80 characters.

ACTION Reenter the append sub-command using less characters, or break the edited line into two lines and then reenter the append sub-command.

4726 MESSAGE **REDO subcommand is followed by unallowed characters (SDERR 4726)**

CAUSE The issued REDO sub-commmand was followed by other characters.

ACTION Some REDO sub-commands must not be accompanied by other characters. Those sub-commands are: break, execute, exit, help, and list. Reenter the sub-command by itself.

4730 MESSAGE **Move forward/backward command ca**n only have a number following it (SDERR 4730)

CAUSE The REDO sub-command is followed by a non-numeric character.

ACTION The forward/backward REDO sub-command must be followed by nothing, or by a number. Reenter the modified sub-command.

4731 MESSAGE **Error occurred while forming the upshift table (SDERR 4731)**

CAUSE 1. The Native Language ID retrieved from the message catalog may be invalid. 2. A Native Language intrinsic error has occurred.

ACTION 1. Change the message catalog to reflect the correct Native Language ID before attempting to run SDCDE again. 2. Verify that the Native Language ID which was retrieved from the message catalog is valid. If not, change it, and rerun SDCDE, otherwise, notify the System Manager.

4732 MESSAGE **Unable to convert SD date into NL date. Error numbererror-number(SDERR 4732)**

CAUSE A Native Language intrinsic error has occurred.

ACTION Notify the System Manager.

4733 MESSAGE **Unable to convert SD time into NL time. Error numbererror-number(SDERR 4733)**

CAUSE A Native Language intrinsic error has occurred.

ACTION Notify the System Manager.

4734 MESSAGE **MPE command execution error.Error code: error-code (SDERR4734)**

CAUSE The MPE command execution resulted in an error.

ACTION Determine the error using the given MPE error code. Fix the command using REDO, or by reentering the command.

4735 MESSAGE **No such MPE command. Error code: error-code (SDERR 4735)**

CAUSE The MPE command does not exist or is unrecognizable.

ACTION Determine the error using the given MPE error code. Fix the command using REDO, or by reentering the command.

4740 MESSAGE **No help on subject help-subject. Displaying general helpinstead (SDERR 4740)**

CAUSE The HELP command parameter is not recognized by the SDCDE help facility.

ACTION The general help messages will be displayed instead. If a specific subject is desired, qualify the HELP command with a valid subject using REDO, or by reentering the command.

SDCDE Operational Warning Messages

4741 MESSAGE **Name truncated while adding prefix to long-name (SDWARN 4741)**

CAUSE The length of the given COBOL name, after adding the prefix to it exceeds 61 characters.

ACTION None necessary. The name is automatically truncated to 61 characters.

4742 MESSAGE **Name truncated while adding suffix to long-name (SDWARN 4742)**

CAUSE The length of the given COBOL name, after adding the suffix to it, exceeds 61 characters.

ACTION None necessary. The name is automatically truncated to 61 characters.

4743 MESSAGE **Invalid-name contains illegal characters (SDWARN 4743)**

CAUSE The given COBOL name contains illegal characters.

ACTION None necessary. The illegal characters are replaced with a hyphen, '-'. Check the manual for a list of illegal characters.

4744 MESSAGE **Invalid-name contains a leading hyphen (SDWARN 4744)**

CAUSE The given COBOL name contains a leading hyphen.

ACTION None necessary. The leading hyphen, '-', will be automatically prefixed by the following string: 'X-'.

4745 MESSAGE **Invalid-name contains a trailing hyphen (SDWARN 4745)**

CAUSE The given COBOL name contains a trailing hyphen.

ACTION None necessary. The trailing hyphen, '-', will be automatically suffixed by the following string: '-X'.

4750 MESSAGE **Invalid-name is a COBOL reserved word (SDWARN 4750)**

CAUSE The given name is a COBOL reserved word.

ACTION None necessary. The name will be automatically suffixed by the following string: '-X'.

4751 MESSAGE **Illegal name above is transformed into transformed-name (SDWARN4751)**

CAUSE Due to a previous error, the name has been transformed into the given valid COBOL name.

ACTION None necessary. This is just an informative message.

4752 MESSAGE **Module line number too big (>= 1000000) (SDWARN 4752)**

CAUSE The module number of the next line (which is automatically generated in SDCDE) exceeds the 6 digit limit, 999999.

ACTION None necessary. SDCDE will prompt you for the name of the module into which the source generation will continue.

4753 MESSAGE **IMAGE-DATABASE data-base contains no data set. No sourcegenerated (SDWARN 4753)**

CAUSE No IMAGE-DATABASE contains IMAGE-DATASET relationships were found in the dictionary for the given database.

ACTION No source will be generated for the database until the relationship(s) are added to the dictionary.

4754 MESSAGE **IMAGE-DATASET data-set contains no RECORD. No source generated**

(SDWARN 4754)

CAUSE No IMAGE-DATASET contains RECORD relationships were found in the dictionary for the given data set.

ACTION No source will be generated for the data set until the relationship(s) are added to the dictionary.

4755 MESSAGE **IMAGE-DATASET data-set has no search item (SDWARN 4755)**

CAUSE No IMAGE-DATASET key ELEMENT relationships were found in the dictionary for the given data set if the set is a master set. If it is a detailed set, no five way chain relationships were found in the dictionary.

ACTION No search item constants will be generated for the data set until the relationship(s) are added to the dictionary.

4760 MESSAGE **Data-set has an illegal data set type. Defaulting to DETAIL(SDWARN 4760)**

CAUSE The image-dataset-type attribute retrieved from the dictionary does not contain a valid data set type.

ACTION The default value, DETAIL, is assigned if the image-dataset-type attribute is invalid. The image-dataset-type attribute must be updated in the dictionary to one of the following valid values: AUTOMATIC, MASTER, or DETAIL.

4761 MESSAGE **Path for search-item data set has a blank search item (SDWARN4761)**

CAUSE In the IMAGE-DATASET, ELEMENT, ELEMENT, IMAGE-DATASET, IMAGE-DATABASE chains relationship, the search item (the first ELEMENT) is a blank entity.

ACTION No search item constant will be generated for the given data set. If one is required, a non-blank search item must be specified in the IMAGE-DATASET, ELEMENT, ELEMENT, IMAGE-DATASET, IMAGE-DATABASE chains relationship.

4762 MESSAGE **FORMSFILE forms-file contains no FORM. No source generated(SDWARN 4762)**

CAUSE No FORMSFILE contains FORM relationships were found in the dictionary for the given forms file.

ACTION No source will be generated for the forms file until the relationship(s) are added to the dictionary.

4763 MESSAGE **FORM form contains no ELEMENTS. No source generated (SDWARN4763)**

CAUSE No FORM contains ELEMENT relationships were found in the dictionary for the given form.

ACTION No source will be generated for the form until the relationship(s) are added to the dictionary.

4764 MESSAGE **FILE file contains no RECORD. No source generated (SDWARN 4764)**

CAUSE No FILE contains RECORD relationships were found in the dictionary for the given file.

ACTION No source will be generated for the file until the relationship(s) are added to the dictionary.

4765 MESSAGE **FILE file uses multiple DEVICE-CLASSES. No dev-cl name incl inASSIGN clause (SDWARN 4765)**

CAUSE Multiple FILE uses DEVICE-CLASS relationships were found for the given file.

ACTION The ASSIGN clause will not include any device name. If one is desired, all but one FILE uses DEVICE-CLASS relationship must be deleted from the dictionary for the given file.

4770 MESSAGE **Relative file cannot produce SD statement. Generating FD(SDWARN 4770)**

CAUSE The SD option was chosen at the FD/SD prompt, when the file being defined in the file section. Relative files can only be defined using the FD statement.

ACTION The FD statement will automatically be used in the file section definition. No action required by user.

4771 MESSAGE **File size cannot be negative (SDWARN 4771)**

CAUSE The file-size attribute of the FILE entity is negative.

ACTION No file size will be included in the SELECT statement. The file-size attribute of the FILE entity should be modified to a non-negative integer value.

4772 MESSAGE **Inconsistent blocking min/max values (SDWARN 4772)**

CAUSE 1. The blocking-min attribute of the FILE entity is a negative number. 2. The blocking-max attribute of the FILE entity is a negative number. 3. The blocking-max attribute is positive, and is less than the blocking-min attribute.

ACTION 1. No BLOCK CONTAINS clause will be generated. The blocking-min attribute of the FILE entity should be modified to a non-negative integer value. 2. No BLOCK CONTAINS clause will be generated. The blocking-max attribute of the FILE entity should be modified to a non-negative integer value. 3. A BLOCK CONTAINS clause will be generated, however, the blocking-min value will be used as the blocking-max value. The blocking-min and/or blocking-max attributes should be modified so that the blocking-max is greater than the blocking-min.

4773 MESSAGE **Inconsistent min/max record size values (SDWARN 4773)**

CAUSE 1. The min-record-size attribute of the FILE entity is a negative number. 2. The max-record-size attribute of the FILE entity is a negative number. 3. The max-record-size attribute is positive, and is less than the min-record-size attribute.

ACTION 1. No RECORD CONTAINS clause will be generated. The min-record-size attribute of the FILE entity should be modified to a non-negative integer value. 2. No RECORD CONTAINS clause will be generated. The max-record-size attribute of the FILE entity should be modified to a non-negative integer value. 3. A RECORD CONTAINS clause will be generated, however, the min-record-size will be used as the max-record-size. The min-record-size and/or max-record-size attributes should be modified so that the max-record-size is greater than the min-record-size.

4774 MESSAGE **Formal file name contains special char's. Name transformed(SDWARN 4774)**

CAUSE The first eight characters of the formal FILE name contain invalid characters.

ACTION The formal name used in the ASSIGN statement will be transformed to a valid name by altering each invalid character to a 'x'. The following are valid characters: a...z, A...Z, 0...9. The formal name of the FILE entity could be altered to contain only valid characters.

4775 MESSAGE **File lockword contains special char's. Lockword transformed(SDWARN 4775)**

CAUSE The lockword attribute of the FILE entity contains invalid characters.

ACTION The lockword will be transformed to a valid lockword by altering each invalid character to a 'x'. The following are valid lockword characters: a...z, A...Z, 0...9. The lockword attribute of the FILE entity should be altered to contain a valid lockword.

4780 MESSAGE **File rec size is zero or negative. Generated dummy record(SDWARN 4780)**

CAUSE The max-record-size attribute of the FILE entity is zero or negative.

ACTION The record description for the FD statement will default to PIC X(2). The max-record-size attribute of the FILE entity should be modified to a positive integer value.

4781 MESSAGE **KSAMFILE ksamfile contains no RECORD. No source generated(SDWARN 4781)**

CAUSE No KSAMFILE contains RECORD relationships were found in the dictionary for the given ksam file.

ACTION No source will be generated for the ksam file until the relationship(s) are added to the dictionary.

4782 MESSAGE **KSAMFILE ksamfile has no key. No key name generated (SDWARN4782)**

CAUSE No KSAMFILE key ELEMENT relationships were found in the dictionary for the given ksam file.

ACTION No RECORD clause(s) in the SELECT section will be generated for the KSAM file until the relationship(s) are added to the dictionary.

4783 MESSAGE **Formal KSAM file name contains special char's. Name transformed(SDWARN 4783)**

CAUSE The first 8 characters of the formal KSAMFILE name contain invalid characters.

ACTION The formal KSAMFILE name used in the ASSIGN clause will be transformed to a valid name by altering each invalid character to a 'x'. The following are valid characters are: a...z, A...Z, 0...9. The formal name of the KSAMFILE entity could be altered to contain only valid characters.

4784 MESSAGE **Boolean not a valid cobol element type. PIC X used! (SDWARN4784)**

CAUSE The element-type attribute retrieved from the dictionary has a value of "B" , boolean. This is not a valid COBOL type.

ACTION The PICTURE clause is generated with a single alphanumeric character, PIC X.

4785 MESSAGE **Floating point not a valid cobol element type. PIC X used!(SDWARN 4785)**

CAUSE The element-type attribute retrieved from the dictionary has a value of "R, D, or E" , floating point. This is not a valid COBOL type.

ACTION The PICTURE clause is generated with a single alphanumeric character, PIC X.

4790 MESSAGE **String not a valid cobol element type. PIC X used! (SDWARN4790)**

CAUSE The element-type attribute retrieved from the dictionary has a value of "S" , string. This is not a valid COBOL type.

ACTION The PICTURE clause is generated with a single alphanumeric character, PIC X.

4791 MESSAGE **Invalid edit mask attribute. Generating edit mask! (SDWARN4791)**

CAUSE The edit-mask attribute retrieved from the dictionary contains invalid characters, and is not a valid COBOL edit mask.

ACTION The edit mask will be generated as if no edit-mask attribute exists in the dictionary. If a special edit mask is required, the dictionary edit-mask attribute must be updated.

4792 MESSAGE **Edit mask too long. Remaining characters truncated! (SDWARN4792)**

CAUSE The edit-mask attribute contains more than 30 characters.

ACTION All characters following the 30th character are truncated, and the remaining 30 characters are validated.

4793 MESSAGE **Sub-elements' total length exceeds parent-element's length(SDWARN 4793)**

CAUSE 1. The total length of all sub-elements is greater than the length of their parent element. 2. The total length of all sub-elements may not be greater than the length of their parent element, but the byte-offset attribute causes the sub-elements to b e placed such that their adjusted length will be greater than the length of their parent element.

ACTION 1. The parent's length is assumed to be valid, and generation continues as if no error had occurred, but the data definition will cause compile warnings. The parent's length should be adjusted to include all child elements, or the child elements' length should be adjusted to fit into the parent element. 2. The parent's length is assumed to be valid, and generation continues as if no error had occurred, but the data definition will cause compile warnings. The parent's length should be adjusted to include all child elements with their current byte offsets, or the child element byte offsets should be adjusted to fit into the parent element.

4794 MESSAGE **Redefining-element's length exceeds redefined entity's length(SDWARN 4794)**

CAUSE 1. The length of the redefining element or record is greater than the length of the redefined element or record. This can happen with the ELEMENT redefines ELEMENT or RECORD redefines RECORD relationship. 2. The length of the redefining element may be greater than the length of the redefined element or record if a forced redefine occurs. The byte-offset attribute may force an element to exceed the length of its parent element or record.

ACTION 1. The length of either the redefining element or record, or the redefined element or record, must be updated to match the length of the other element or record. 2. Either the byte-offset attribute or the redefined element or records length must be updated in the dictionary.

4795 MESSAGE **Multiple ELEMENT references ELEMENT occurrences forfirst-element (SDWARN 4795)**

CAUSE The element which contains the back reference element type has more than one ELEMENT references ELEMENT relationship occurrences in the dictionary.

ACTION The ELEMENT references ELEMENT relationship with the lowest relationship-position attribute value will be used as the back referenced element. If this is not the required back reference, the undesired ELEMENT references ELEMENT relationships must be deleted from the dictionary.

4799 MESSAGE **Unable to get condition value variable for condition-name(SDWARN 4799)**

CAUSE 1. No CONDITION-VALUE variable attribute exists for the ELEMENT contains CONDITION-NAME relationship. 2. The CONDITION-VALUE variable attribute has a length of zero.

ACTION 1. No level-88 definition will be generated for the relationship until a CONDITION-

VALUE variable attribute is created. 2. No level-88 definition will be generated for the relationship until a CONDITION-VALUE variable attribute length is greater than zero.

4800 MESSAGE **Unable to get var attribute. No default used! (SDWARN 4800)**

CAUSE A System Dictionary error occurred while retrieving the default attribute for an ELEMENT.

ACTION No VALUE clause is generated for the element. If a serious System Dictionary error is suspected, notify the Dictionary Administrator.

4801 MESSAGE **Unable to get var attribute. Created edit mask! (SDWARN 4801)**

CAUSE A System Dictionary error occurred while retrieving the edit-mask attribute for an ELEMENT.

ACTION The edit mask is generated using other attribute values. If a serious System Dictionary error is suspected, notify the Dictionary Administrator.

4802 MESSAGE **Invalid sign attribute. Sign clause not created! (SDWARN 4802)**

CAUSE The sign attribute retrieved from the dictionary contains an invalid value.

ACTION No SIGN clause is created for the element. The sign attribute should be updated to one of the following valid values: LS, LO, TS, TO, or <blank>.

4803 MESSAGE **RECORD record contains no ELEMENT (SDWARN 4803)**

CAUSE No RECORD contains ELEMENT relationships were found for the given record.

ACTION The record will not be further defined, thus the definition will produce compiler errors. In order to generate the record correctly, RECORD contains ELEMENT relationship(s) should be added to the dictionary.

4804 MESSAGE **Non-numeric default value too long. Truncated! (SDWARN 4804)**

CAUSE The default variable attribute contains more than 132 characters.

ACTION The first 132 characters will be used in the VALUE clause. The default variable attribute should be updated in the dictionary.

4805 MESSAGE **Numeric default value too long. None generated! (SDWARN 4805)**

CAUSE The default variable attribute contains more than 20 numbers and characters.

ACTION No VALUE clause will be generated for the element. The default variable attribute should be updated in the dictionary.

4806 MESSAGE **Justify not allowed when parent contains level-88 (SDWARN 4806)**

CAUSE The JUSTIFY attribute is not allowed for an ELEMENT that is directly indirectly contained by an ELEMENT that contains a CONDITION-NAME.

ACTION Either the JUSTIFY attribute must be false, or the ELEMENT contains CONDITION-NAME relationship must be deleted.

4807 MESSAGE **Synchronize not allowed when parent contains level-88 (SDWARN4807)**

CAUSE The SYNCHRONIZE attribute is not allowed for an ELEMENT that is directly or indirectly contained by an ELEMENT that contains a CONDITION-NAME.

ACTION Either the SYNCHRONIZE attribute must be false, or the ELEMENT contains CON-

DITION-NAME relationship must be deleted.

4808 MESSAGE **Usage is COMP not allowed when parent contains level-88 (SDWARN4808)**

CAUSE The ELEMENT-TYPE attribute values I, J, or K are not allowed for an ELEMENT that directly or indirectly contained by an ELEMENT that contains a CONDITION-NAME.

ACTION Either the ELEMENT-TYPE attribute must be modified, or the ELEMENT contains CONDITION-NAME relationship must be deleted.

4809 MESSAGE **Usage is COMP3 not allowed when parent contains level-88 (SDWARN4809)**

CAUSE The ELEMENT-TYPE P is not allowed for an ELEMENT that is directly or indirectly contained by an ELEMENT that contains a CONDITION-NAME.

ACTION Either the ELEMENT-TYPE attribute must be modified, or the ELEMENT contains CONDITION-NAME relationship must be deleted.

4810 MESSAGE **Invalid numeric default value. None generated! (SDWARN 4810)**

CAUSE The default variable attribute contains more than 18 numbers, or more than 2 edit characters.

ACTION No VALUE clause will be generated for the element. The default variable attribute should be updated in the dictionary.

4811 MESSAGE **Numeric picture exceeds 18 digits. PIC X used! (SDWARN 4811)**

CAUSE The generated edit mask allows more than 18 digits. The byte-length attribute must not exceed 18 for numeric elements.

ACTION The PICTURE clause is generated with a single alpha-numeric character, PIC X. The byte-length attribute should be updated to a positive value not exceeding 18, in the dictionary.

4812 MESSAGE **Expected byte length of 2, 4, or 8. PIC X used! (SDWARN 4812)**

CAUSE The byte length attribute must be 2, 4, or 8 for elements of type I, I+, J, J+, K, and K+.

ACTION The PICTURE clause is generated with a single alpha-numeric character, PIC X. The byte-length attribute should be updated in the dictionary to one of the following values: 2, 4, or 8.

SDCDE Fatal Error Messages

4813 MESSAGE **SDCDE version incompatible with that of SD intrinsic (SDERR4813)**

CAUSE The program's version number is different from the intrinsic's version number.

ACTION Notify the Dictionary Administrator.

4814 MESSAGE **Program aborts due to too many errors (SDERR 4814)**

CAUSE The program detected the number of errors specified through the run-option parameter.

ACTION Check your input and see where the errors have occurred.

4815 MESSAGE **$STDINX open error. Program aborted (SDERR 4815)**

CAUSE An error occurred while opening $STDINX.

ACTION Notify the System Manager.

4820 MESSAGE **Unable to read from input file. Program aborted (SDERR 4820)**

CAUSE A disc error occurred while reading from the program input file.

ACTION Check the file system error message. Notify the System Manager if a serious problem is suspected.

4821 MESSAGE **Premature EOF reached on input file. Program aborted (SDERR4821)**

CAUSE The EOF is found while reading from the input file.

ACTION Edit the SDIN file to correct the problem. Be sure to include the EXIT command before EOF.

4822 MESSAGE **Unable to write to SDLOG. Program aborted (SDERR 4822)**

CAUSE A disc error occurred while writing to SDLOG.

ACTION Check the file system error message. See if the disc is full. Notify the System Manager if a serious error is suspected.

4823 MESSAGE **Unable to close SDLOG. Program aborted (SDERR 4823)**

CAUSE A disc error occurred while closing SDLOG.

ACTION Check the file system error message. Notify the System Manager if a serious error is suspected.

4824 MESSAGE **Unable to write to copylib. Program aborted (SDERR 4824)**

CAUSE A disc error occurred while writing to the copy library.

ACTION Check the file system error message, and make sure that the copylib is not full. Notify the System Manager if a serious error is suspected. If FSERR 0 (END OF FILE), the copylib may contain records that have been deleted. These records are not discarded until an FCOPY with option KEY=0 is used. This is referred to as Compacting a KSAMFILE.

4825 MESSAGE **Unable to find a record by key in copylib. Program aborted(SDERR 4825)**

CAUSE The FFINDBYKEY intrinsic failed while attempting to find a copy library record using a key value.

ACTION Make sure that the copy library has not been corrupted. Notify the System Manager if a serious error is suspected.

4830 MESSAGE **Unable to read from copylib. Program aborted (SDERR 4830)**

CAUSE A disc error occurred while reading from the copy library.

ACTION Check the file system error message. Notify the System Manager if a serious error is suspected.

4831 MESSAGE **Unable to remove records from copylib. Program aborted (SDERR4831)**

CAUSE The FREMOVE intrinsic failed while attempting to remove lines from an copy library module that is being rewritten.

ACTION Notify the System Manager if a serious error is suspected.

4832 MESSAGE **Unable to open temporary file SDTMP. Program aborted (SDERR4832)**

CAUSE A disc error occurred while opening the temporary file SDTMP.

ACTION Check the file system error message. Notify the System Manager if a serious error is suspected.

4833 MESSAGE **Unable to read from temp file SDTMP. Program aborted (SDERR4833)**

CAUSE A disc error occurred while reading from the temporary file SDTMP.

ACTION Check the file system error message. Notify the System Manager if a serious error is suspected.

4834 MESSAGE **Unable to write to temp file SDTMP. Program aborted (SDERR 4834)**

CAUSE A disc error occurred while writing to the temporary file SDTMP.

ACTION Check the file system error message. Notify the System Manager if a serious error is suspected.

4835 MESSAGE **Unable to delete temp file SDTMP. Program aborted (SDERR 4835)**

CAUSE A disc error occurred while deleting the temporary file SDTMP.

ACTION Check the file system error message. Notify the System Manager if a serious error is suspected.

4840 MESSAGE **Unable to open temp file SDREDO. Program aborted (SDERR 4840)**

CAUSE A disc error occurred while opening the temporary file SDREDO.

ACTION Check the file system error message. Notify the System Manager if a serious error is suspected.

4841 MESSAGE **Unable to read from temp file SDREDO. Program aborted (SDERR4841)**

CAUSE A disc error occurred while reading from the temporary file SDREDO.

ACTION Check the file system error message. Notify the System Manager if a serious error is suspected.

4842 MESSAGE **Unable to write to temp file SDREDO. Program aborted (SDERR4842)**

CAUSE A disc error occurred while writing to the temporary file SDREDO.

ACTION Check the file system error message. Notify the System Manager if a serious error is suspected.

4843 MESSAGE **Unable to delete temp file SDREDO. Program aborted (SDERR 4843)**

CAUSE A disc error occurred while deleting the temporary file SDREDO.

ACTION Check the file system error message. Notify the System Manager if a serious error is suspected.

4844 MESSAGE **Unable to open temp file SDTEXT. Program aborted (SDERR 4844)**

CAUSE A disc error occurred while opening the temporary file SDTEXT.

ACTION Check the file system error message. Notify the System Manager if a serious error is suspected.

4845 MESSAGE **Unable to reset temp file SDTEXT. Program aborted (SDERR 4845)**

CAUSE A disc error occurred while resetting the temporary file SDTEXT.

ACTION Check the file system error message. Notify the System Manager if a serious error is suspected.

4850 MESSAGE **Unable to write to temp file SDTEXT. Program aborted (SDERR4850)**

CAUSE A disc error occurred while writing to the temporary file SDTEXT.

ACTION Check the file system error message. Notify the System Manager if a serious error is suspected.

4851 MESSAGE **Unable to read from temp file SDTEXT. Program aborted (SDERR4851)**

CAUSE A disc error occurred while reading from the temporary file SDTEXT.

ACTION Check the file system error message. Notify the System Manager if a serious error is suspected.

4852 MESSAGE **Unable to delete temp file SDTEXT. Program aborted (SDERR 4852)**

CAUSE A disc error occurred while deleting the temporary file SDTEXT.

ACTION Check the file system error message. Notify the System Manager if a serious error is suspected.

4853 MESSAGE **Unable to open temp file SDRECUR. Program aborted (SDERR 4853)**

CAUSE A disc error occurred while opening the temporary file SDRECUR.

ACTION Check the file system error message. Notify the System Manager if a serious error is suspected.

4854 MESSAGE **Unable to read from temp file SDRECUR. Program aborted (SDERR4854)**

CAUSE A disc error occurred while reading from the temporary file SDRECUR.

ACTION Check the file system error message. Notify the System Manager if a serious error is suspected.

4855 MESSAGE **Unable to write to temp file SDRECUR. Program aborted (SDERR4855)**

CAUSE A disc error occurred while writing to the temporary file SDRECUR.

ACTION Check the file system error message. Notify the System Manager if a serious error is suspected.

4860 MESSAGE **Unable to delete temp file SDRECUR. Program aborted (SDERR 4860)**

CAUSE A disc error occurred while deleting the temporary file SDRECUR.

ACTION Check the file system error message. Notify the System Manager if a serious error is suspected.

4861 MESSAGE **Unable to open temp file SDRELLST. Program aborted (SDERR 4861)**

CAUSE A disc error occurred while opening the temporary file SDRELLST.

ACTION Check the file system error message. Notify the System Manager if a serious error is suspected.

4862 MESSAGE **Unable to reset temp file SDRELLST. Program aborted (SDERR 4862)**

CAUSE A disc error occurred while resetting the temporary file SDRELLST.

ACTION Check the file system error message. Notify the System Manager if a serious error is suspected.

4863 MESSAGE **Unable to write temp file SDRELLST. Program aborted (SDERR 4863)**

CAUSE A disc error occurred while writing to the temporary file SDRELLST.

ACTION Check the file system error message. Notify the System Manager if a serious error is suspected.

4864 MESSAGE **Unable to read from temp file SDRELLST. Program aborted (SDERR4864)**

CAUSE A disc error occurred while reading from the temporary file SDRELLST.

ACTION Check the file system error message. Notify the System Manager if a serious error is suspected.

4865 MESSAGE **Unable to delete temp file SDRELLST. Program aborted (SDERR4865)**

CAUSE A disc error occurred while deleting the temporary file SDRELLST.

ACTION Check the file system error message. Notify the System Manager if a serious error is suspected.

4870 MESSAGE **Unable to set EOF on message buffer file. Program aborted(SDERR 4870)**

CAUSE The FCONTROL intrinsic failed on the message buffer file (SDREDO).

ACTION Notify the System Manager.

4871 MESSAGE **Unable to read from message buffer file. Program aborted (SDERR4871)**

CAUSE The FREAD intrinsic failed while reading from the message buffer file (SDREDO).

ACTION Notify the System Manager.

4872 MESSAGE **Source line formatting algorithm failed (SDERR 4872)**

CAUSE The algorithm used to format each line of the generated definition has found a situation that it was not designed to handle.

ACTION Notify the Dictionary Administrator.

4873 MESSAGE **Exceeded max number of tries for a valid password. Prog aborted(SDERR 4873)**

CAUSE A valid SD password was not entered within the maximum number of times allowed, while in job mode.

ACTION Correct the password and re-run the job.

4880 MESSAGE **Unable to open temp file SDCNST. Program aborted (SDERR 4880)**

CAUSE A disc error occurred while opening the temporary file SDCNST.

ACTION Check the file system error message. Notify the System Manager if a serious error is suspected.

4881 MESSAGE **Unable to reset temp file SDCNST. Program aborted (SDERR 4881)**

CAUSE A disc error occurred while resetting the temporary file SDCNST.

ACTION Check the file system error message. Notify the System Manager if a serious error is suspected.

4882 MESSAGE **Unable to write temp file SDCNST. Program aborted (SDERR 4882)**

CAUSE A disc error occurred while writing to the temporary file SDCNST.

ACTION Check the file system error message. Notify the System Manager if a serious error is

suspected.

4883 MESSAGE **Unable to read from temp file SDCNST. Program aborted (SDERR4883)**

CAUSE A disc error occurred while reading from the temporary file SDCNST.

ACTION Check the file system error message. Notify the System Manager if a serious error is suspected.

4884 MESSAGE **Unable to delete temp file SDCNST. Program aborted (SDERR 4884)**

CAUSE A disc error occurred while deleting the temporary file SDCNST.

ACTION Check the file system error message. Notify the System Manager if a serious problem is suspected.

None MESSAGE **Message catalog (SDCAT.PUB.SYS) open error.**

CAUSE SDCAT.PUB.SYS message catalog does not exist.

ACTION Notify Dictionary Administrator of the problem.

None MESSAGE **Message catalog (SDCAT.PUB.SYS) access error.**

CAUSE SDDBD could not access SDCAT.PUB.SYS.

ACTION Check and see if SDCAT.PUB.SYS is being exclusively accessed by some other users. If a serious problem is suspected, notify System Manager and Dictionary Administrator.

None MESSAGE **Message catalog (SDCAT.PUB.SYS) illegally formatted.**

CAUSE SDCAT.PUB.SYS is not in accordance with the NL message catalog format.

ACTION Notify Dictionary Administrator of this problem.

None MESSAGE **Message catalog (SDCAT.PUB.SYS) internal error**

CAUSE SDCAT.PUB.SYS contains an internal error.

ACTION Notify Dictionary Administrator and System Manager of this problem.

None MESSAGE **Message catalog access error: set = set-no, msg = msg-no, err =err-no**

CAUSE An error occurred while attempting to access an SDDBD message. The message set number is **sss**, message line number is **mmm**, and the NL error code is **eee**.

ACTION Check the message catalog (SDCAT.PUB.SYS) and see if the message is missing. If a serious problem is suspected, notify System Manager.

None MESSAGE **Error while writing to $STDLIST. Program is aborted.**

CAUSE An error is detected while writing to $STDLIST.

ACTION If the SDDLIST = clause is used at the RUN command, see if the file equated to $STDLIST is filled up. If a serious problem is suspected, notify System Manager.

B SDCDE Command Abbreviations

SDCDE accepts abbreviations for the command and sub-command keywords. The following table lists each SDCDE keyword and its corresponding abbreviation.

Table 2: Command Abbreviations

Keyword	Abbreviation
COBEDIT	CE
COMMENT	COM
COPYLIB	CL
NAME	N
KEY-FILE	KF
UPDATE-DICT	UD
ON	ON
OFF	OFF
OPEN-MODE	OM
APPEND	AP
REPLACE	RE
DEFINE	DEF
DICTIONARY	DICT
SCOPE	S
PASSWORD	P
DOMAIN	D
VERSION	V
STATUS	STAT
TEST	TEST
PRODUCTION	PROD
ARCHIVAL	ARCH
OPEN-MODE	OM
READ-ONLY	RO
SHARED-UPDATE	SU
EXCLUSIVE-UPDATE	EU
NAME-MODE	NM
EXTERNAL	EXT
INTERNAL	INT
EXIT	E
GENERATE	GEN
ENTITY-TYPE	ET
HELP	H

Table 2: Command Abbreviations

Keyword	Abbreviation
LIST MODULE	L MOD
OPTIONS ALIAS NONE COBOL IMAGE STANDARD VPLUS EDIT-MASK ON OFF QUALIFY ELEM-QUALIFY NONE PREFIX SUFFIX CONSTANTS SELECT-FILE COMMON-STORE SECONDARY-REC ECHO COMMENT	OP A NO COB IM STD VP EM ON OFF Q EQ NO PRE SUF CON SF CS SR EC COM
REDO	REDO
RESET	R
SHOW	SH
STANDARD-PARMS IMAGE VPLUS KSAM	SP IM VP KS

C SDCDE PICTURE Clause Translation

Generating the Edit Mask from Attributes

The generation of the PICTURE clause for each data element depends on whether or not you request the use of the edit-mask. If you do not request that the edit-mask be used, Table C-1 shows how SDCDE generates the PICTURE clause.

In Table C-1, "d" designates the display-length attribute value, "b" the byte-length attribute value, and "e" the decimal attribute value.

Table 3:

System Dictionary Element Type	COBOL PICTURE Clause
X , U	X(d)
9 , 9+	9(d)
Z+	If e > 0, then 9(b - e)V9(e) If e = 0, then 9(b) If e < 0, SDCDE issues a warning and generates the PICTURE clause as if e = 0

Z with sign separate	If e > 0, then
	S9(b-e-1)V9(e) SIGN LEADING SEPARATE or
	S9(b-e-1)V9(e) SIGN TRAILING SEPARATE
	If e = 0, then
	S9(b-1) SIGN LEADING SEPARATE or
	S9(b-1) SIGN TRAILING SEPARATE
	If e < 0, SDCDE issues a warning and generates the PICTURE clause as if e = 0

Z with sign over-punch	If e > 0, then
	S9(b-e)V9(e) SIGN LEADING or SIGN TRAILING
	If e = 0, then
	S9(b) SIGN LEADING or SIGN TRAILING
	If e < 0, SDCDE issues a warning and generates the PICTURE clause as if e = 0

P	If e = 0, then S9(n) COMP-3, where n = 2b-1, if d >= 2b-1 n = 2b-2, if d <= 2b-2 If e > 0, then S9(n)V9(e) COMP-3, where n = 2b-e-1, if d >= 2b n = 2b-e-2, if d <= 2b-1 If e < 0, SDCDE issues a warning and generates the PICTURE clause as if e = 0
P+	If e = 0, then 9(n) COMP-3, where n = 2b-1, if d >= 2b-1 n = 2b-2, if d <= 2b-2 If e > 0, then 9(n)V9(e) COMP-3, where n = 2b-e-1, if d >= 2b n = 2b-e-2, if d <= 2b-1 If e < 0, SDCDE issues a warning and generates the PICTURE clause as if e = 0

Table 3:

System Dictionary Element Type	COBOL PICTURE Clause
I , J , K	If e = 0, then S9(n) COMP where b d n --- ------- --- 2 d < 4 d d > 4 4 4 d < 5 5 5 <= d <= 9 d d > 9 9 8 d < 10 10 10 <= d <= 18 d d > 18 18 If b does not equal 2, 4, or 8 then SDCDE issues a warning message and generates the PICTURE clause as X(d) If e > 0, then S9(n)V9(e) COMP where b d n --- ------- --- 2 d <= 5 d-e-1 d > 5 4-e 4 d < 6 5-e 6 <= d <= 10 d-e-1 d > 10 9-e 8 d < 11 10-e 11 <= d <= 19 d-e-1 d > 19 18-e If b does not equal 2, 4, or 8 then SDCDE issues a warning message and generates the PICTURE clause as X(d) If e < 0, SDCDE issues a warning and generates the PICTURE clause as if e = 0
I+ , J+ , K+	If e = 0, then 9(n) COMP where (see I , J , K above for the value of n) If e > 0, then 9(n)V9(e) COMP where If e < 0, SDCDE issues a warning and generates the PICTURE clause as if e = 0
R , E , B , S , D	SDCDE issues a warning message and generates the PICTURE clause as X(b)
*	SDCDE searches ELEMENT references ELEMENT relationship to find the element definition

Generating the PICTURE Clause from the Edit Mask

If you request generation of the PICTURE clause from the edit-mask, all other attributes pertaining to the clause are ignored. Only if the element's type cannot be determined from the edit-mask is the element-type attribute used to determine if the value is alphanumeric or numeric.

If no edit-mask attribute exists, the PICTURE is determined using other attribute data as shown in Table C-1.

SDCDE uses the synchronize and justify attributes to generate the corresponding SYNCHRONIZE LEFT and JUSTIFIED RIGHT clauses, regardless of whether the PICTURE clause is generated from the edit-mask attribute or not.

In order to generate the PICTURE clause from the edit-mask, some characters must be translated into COBOL edit characters. Table C-2 maps the translation of each System Dictionary edit-mask character to its corresponding COBOL edit mask character.

Table 4: Edit-Mask Translation

SYSTEM DICTIONARY EDIT MASK CHARACTER	COBOL EDIT MASK CHARACTER
^	X if type is X or U, otherwise 9
(blank)	B
DR	DB
!	Not a COBOL edit character
.	.
,	,
/	/
B	B
0	0
$	$
*	*
Z	Z
+	+
-	-
CR	CR
DB	DB
A	A
X	X
9	9
S	S
V	V
P	P
All others	No other characters are allowed

NOTE The currency sign would not be $ if the CURRENCY SIGN IS clause is used.

In addition to the above, the repetition characters are also considered valid COBOL edit characters. Thus, any numbers that are enclosed in parentheses (X(15)) are considered valid.

D Glossary

This appendix provides a glossary of System Dictionary terms.

Access - The right to read or manipulate a dictionary domain or occurrence.

Access Rights - The rights of a scope to read or manipulate a domain or occurrence, as determined by whether that scope is the owner of the item, or is just associated with it.

Alias Name - Different names associated with different external subsystem uses of an occurrence, including a difference in programming syntax (e.g. the use of the underscore (_) instead of the hyphen (-) in names).

Association - An explicit access assigned between a scope and a domain, entity, or relationship, which has been granted to that scope by the owner scope of the domain, entity, or relationship.

Attribute - An object in the dictionary structure that is a piece of information describing an entity type or relationship type.

Attribute Edit - A value or range of values used for determining if input attribute values are valid when creating or modifying an occurrence. Also used to specify the default attribute value to use when an occurrence is created.

Attribute Prompting - A facility that prompts for attribute values whenever you issue a CREATE, MODIFY, or REPORT command without the ATTRIBUTE-LIST parameter. (SDMAIN only)

Attribute Value - The specific information (e.g. text, numbers, etc) assigned to an attribute, describing a particular *occurrence* of an entity type or relationship type.

Binary Relationship - A relationship involving two entities.

Child Entity - The second entity in a relationship.

Command - The SDMAIN-defined name that specifies the action to be taken.

Common Domain - The primary name space for dictionary occurrences. It is provided with System Dictionary, is represented by a blank name, is owned by the core set, has a sensitivity of *Public*, and can never be modified or deleted.

Core Set - A predefined set of entity types, relationship types, relationship classes, attributes, and domains that are provided with System Dictionary. It also includes the scope CORESET, which owns everything in the core set. A second scope, the Dictionary Administrator scope, is included in the core set, and is also owned by the scope CORESET. **DA scope -** See Dictionary Administrator Scope.

DCB - See Dictionary Control Block.

Dictionary Administrator Scope - A special scope provided with the System Dictionary core set, which has unlimited access to all items in the dictionary, and ultimate authority.

Dictionary Control Block - An array of data which contains information about the current status of the dictionary to an intrinsic.

Dictionary Environment - The dictionary environment includes the name of the dictionary that is open, the scope, the open mode, the name mode, the domain, version, and version status that are used for creating and retrieving definitions.

Domain - A name space within the dictionary. See Common Domain and Local Domain.

E-R Model - See Entity-Relationship model.

Entity - An entity is a description of an object in the information network, and belongs to a specific Entity Type.

Entity List - The ordered list of entities that make up a relationship.

Entity Type - An object in the dictionary structure that classifies entity occurrences. Each entity type is further defined by an associated set of attributes.

Entity-Relationship Model - A logical structure that is general enough that it can describe most, if not all, of the information processing done on a computer network. The entity-relationship model is composed of entity types, relationship types, relationship classes and attributes.

Environment - 1) The computer system hardware and software required for the operation of System Dictionary.

Extended Set - The user-created set of structure definitions within the dictionary; an extension of the Core Set.

External Name - One of two names (see also Internal Name) assigned to every item in the dictionary. It is a customizable and localizable reference that is intended for dictionary end users.

Homonym - The same name used for conceptually different entity occurrences of the same entity type.

Internal Name - One of two names (see also External Name) assigned to every item in the dictionary. An internal name is not changeable, and is intended for use by software products used with System Dictionary which rely on specific names for identification purposes.

Internal Number - An identification number automatically assigned to all dictionary components when they are created. These numbers may be read from the Status array (parameter) of intrinsics used for creation and retrieval of dictionary components and, when used, can greatly increase the efficiency and speed of some dictionary operations.

Keyword-Clause - A keyword clause can be either a single keyword or a keyword followed by an equal sign (=) that is followed by either nothing, a single value, or a list of values separated by commas. The keywords are SDMAIN-defined, while their values are either SDMAIN defined or user-defined. Keyword clauses are separated by semicolons.

List Terminator - A semicolon (**;**) that indicates to the intrinsic using a specific list, that there are no more entries in the list.

Local Domain - A user-created name space that separates a set of names, which includes names used for a different purpose. See also Common Domain.

Locking - A process that allows only one user at a time to access the dictionary. System Dictionary provides two types of locking: automatic, which protects individual operations, and manual, which can protect a sequence of operations.

Logging - A process that can automatically create a log of all dictionary transactions, providing a means to repeat those transactions in the event of data loss.

Macro - A user-defined set of commands that you can save in a file and call using macro names. When you call the macro, each defined command is executed in the same way that it would have been had you entered each command individually. (SDMAIN only)

Metadata - Descriptive information about data, but not the data itself. Example: a file card in a library, which contains information about a book, but is not the book itself; an address of a building, which provides information about its location, but is not the location itself.

N-ary Relationship - A relationship that involves **N** entities, where **3<=N<=6** (see also Binary

Relationship).

Name mode - A parameter set while opening the dictionary, used to cause intrinsics to reference either internal or external names when accessing dictionary items.

Name set - A group of names within the dictionary that includes names for any one of the following types of dictionary definitions: domains, versions in the same domain, entity types, relationship classes, attributes, scopes, and entity occurrences of a specific type that are located in the same domain.

Object-Clause - The user-defined name of the object. This is the specific target of the action specified by the command.

Occurrence - A specific instance of an entity or relationship.

Open mode - One of four dictionary operating modes, set when opening the dictionary for use.

Owner scope - A scope that is directly associated with an object in the dictionary and has all rights to it, because the scope has either created that object, or has been given ownership by the scope that created it or previously owned it.

Parent Entity - The first entity in a relationship.

Password - A combination of up to 32 special or alphanumeric characters, and/or blanks used for user identification purposes to limit access to data or objects within the dictionary.

Primary Name - The principal name of an entity, not a synonym, that is initially assigned when the entity is created. Whenever an entity name is returned by System Dictionary, the primary name is returned.

Relationship - A logically connected, ordered series of two to six entities, which belongs to a specific Relationship Type.

Relationship class - The specific class of association or logical connection between the entities in a relationship.

Relationship position - The logical order of a child entity (the second entity in a relationship) relative to all other child entities for the same parent entity (the first entity in a relationship) of the same relationship type.

Relationship type - An object in the dictionary structure that classifies relationship occurrences; a logical connection between entity types specified by a series of two to six ordered entity types and a relationship class. Each relationship type is further defined by an associated set of attributes.

Restructuring - A process similar to compiling. Restructuring incorporates all changes made to the dictionary structure in a single session into the working dictionary, and reformats any dictionary occurrences that are affected by those structure changes.

Scope - A security definition within the dictionary environment that sets the level of access a user has to all objects in the dictionary. It includes up to six scope rights.

Scope Right - One of six specific capabilities associated with a scope. The scope right specifies which dictionary components that scope is allowed to manipulate.

Security - A protection scheme within System Dictionary that limits access to objects in the dictionary to authorized users. The primary elements of dictionary security are scopes In addition, dictionary domains and occurrences each have a sensitivity, which further define their access by a specific scope.

Sensitivity - An access right associated with a dictionary domain or occurrence.

Special Attributes - The set of attributes that are automatically assigned to entity types and relationship types when the types are created. **Status -** Information about the success or failure of an

intrinsic call. The status is returned as the final parameter of intrinsic calls.

Structure - The part of System Dictionary that includes both core set and extended set entity types, relationship types, relationship classes, and attributes.

Subcommand - The SDMAIN-defined name hat specifies the general target of the action.

Synonym - An alternate name for an entity in the dictionary. A synonym must uniquely identify a given entity.

Variable Length Attribute - An attribute whose value must be explicitly defined, and whose length is dependent upon that value. Example: an attribute *description*, whose value is sixty bytes of text. Therefore, the length of the attribute is sixty.

Version - A set of occurrences within a domain, set apart from other sets within the domain.