

SH20-9083-2

Program Product

**DB/DC Data Dictionary
Terminal User's Guide
and Command Reference**

**Program Numbers 5740-XXF
5746-XXC**

IBM

This publication was produced using the
IBM Document Composition Facility
(program number 5748-XX9)
and the master was printed on the
IBM 3800 Printing Subsystem.

Third Edition (July 1979)

This is a major revision of, and makes obsolete, SH20-9083-1, and its technical newsletter, SN20-9263.

This edition applies to Release 3.0 of the DB/DC Data Dictionary, program numbers 5740-XXF and 5746-XXC, and to any subsequent releases and modifications until otherwise indicated in new editions or technical newsletters.

This edition is part of a new library for Release 3.0 of the DB/DC Data Dictionary. Its scope is different from the edition it replaces. It incorporates all technical changes introduced by Release 3.0 of the product. Because the technical changes in this edition are extensive and difficult to localize, they are not marked by vertical bars in the left margin.

Changes are continually made to the information herein; before using this publication in connection with the operations of IBM systems, consult the latest IBM System/370 Bibliography, GC20-0001, for the editions that are applicable and current.

Publications are not stocked at the address given below; requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, P.O. Box 50020, Programming Publishing, San Jose, California, U.S.A. 95150. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

PREFACE

This book is a guide for terminal operators, librarians, and others who use, or direct others to use, the DB/DC Data Dictionary (hereafter referred to as the Dictionary). It is intended as a guide to using the Dictionary online and preparing batch input streams, and as a reference for all Dictionary commands, batch forms, and display forms. It is intended as a companion book to the DB/DC Data Dictionary Applications Guide.

PREREQUISITE PUBLICATIONS

Before using this book, you should read the DB/DC Data Dictionary General Information Manual, GH20-9104, as an introduction to the basic application of the Dictionary and the terminology associated with the Dictionary.

If you intend to use the IBM 3277 Display Systems, to work with the interactive display forms, you should refer to the Operator's Guide for IBM 3270 Information Display Systems, GA27-2742, or equivalent publications for other terminals.

COREQUISITE PUBLICATION

If you need information about how the Dictionary relates to your installation's applications read the DB/DC Data Dictionary Applications Guide, SH20-9173. This book contains extensive examples of the use of commands and forms.

HOW THIS BOOK IS ORGANIZED

- Chapter 1 contains information about Dictionary subjects and categories. The rules for defining subject names are fully discussed. Security considerations are described.
- Chapter 2 contains information about using the Dictionary online with either IMS/VS or CICS/VS. It also contains an introduction to display forms. This introduction combines general information about the display forms with a detailed description of the HEADER forms. Any user of the display forms should be familiar with all the material in this chapter before attempting to use the forms.
- Chapter 3 contains descriptions of all the display forms that a general user will need to use. This chapter is intended to be a reference guide to the forms.
- Chapter 4 describes the structure of a batch input stream, the kind of commands and Dictionary controls that may be included, and the special forms of batch execution for bulk input.
- Chapter 5 explains how to use the batch forms input facility to enter segment definitions, field definitions, and text data into the Dictionary.
- Chapter 6 summarizes the Dictionary command language and gives detailed reference information for each command.
- Appendixes A and B give information you will need to use Dictionary commands.
- Appendix C summarizes and explains all the Dictionary system messages, including those you may receive while working with the interactive display forms facility.

An index is included at the end of the book.

For convenient information about the Dictionary or DL/I terms, you may refer to the glossaries in the DB/DC Data Dictionary General Information Manual, GH20-9104, or in the DB/DC Data Dictionary Applications Guide, SH20-9073, or to the IMS/VS Master Index and Glossary, SH20-9085.

RELATED PUBLICATIONS

If you use the batch forms input facility, special preprinted forms are available to aid you in entering your definitions:

- DB/DC Data Dictionary Field Definition Form, GX26-3716
- DB/DC Data Dictionary Field Definition Form for PL/I Data, GX26-3725
- DB/DC Data Dictionary Segment Definition Form, GX26-3717
- DB/DC Data Dictionary Text Data Form, GX26-3718

Other publications referred to in this book are:

- DB/DC Data Dictionary Administration and Customization Guide
- DOS/VS Sort/Merge Reference Guide, SC33-4035
- OS/VS Sort/Merge Reference Guide, SC33-4028
- OS/VS JCL Reference, GC28-0618
- DL/I DOS/VS Operator's Reference Manual and Messages and Codes, SH12-5414
- IMS/VS System Programming Reference Manual, SH20-9027

NOTATIONAL CONVENTIONS

A uniform system of notation describes the format of Dictionary commands. This notation is not part of the language; it simply provides a basis for describing the structure of the commands.

The command-format illustrations in this book use the following conventions:

- Brackets [] indicate optional parameters.
- Braces { } indicate a choice of entry; unless a default is indicated, you must choose one of the entries.
- Items separated by a vertical bar represent alternative items. No more than one of the items may be selected.
- An ellipsis ... indicates that multiple entries of the type immediately preceding the ellipsis are allowed.
- Other punctuation (parentheses, commas, spaces, etc.) must be entered as shown.
- Uppercase type in **Boldface** indicates the exact characters to be entered. Such items must be entered exactly as illustrated.
- Lowercase type in **Boldface** indicates values to be supplied by the user.
- underscored type indicates a default option. If the parameter is omitted, the underscored value is assumed.

CONTENTS

Chapter 1. Dictionary Subjects and Subject Categories	1
Subject Categories and Subject Names	1
Subject Name	1
Aliases	2
Subject Data	2
Relationships	2
Relationship Data	2
Definitions in Subject Categories	2
Subject Names and Subject Name Rules	4
Subject Name Qualifier Defaults	8
Modifying Installation-defined Default Values	8
Security Considerations	8
Chapter 2. Using the Dictionary Online	9
Using the Dictionary Online with IMS/VS	9
Starting A Session	9
Terminating a Session	10
Output from IMS/VS Online Operations	10
Using the Dictionary Online with CICS/VS	10
Starting a Session	10
Terminating a Session	11
Shutting Down the Dictionary Environment	11
Output from CICS/VS Online Commands	12
Segmented Output	12
Routing of Output	12
Using Display Forms	12
Kinds of Display Forms	13
General Format and Procedures	16
Modes of Operation	16
The Action Area	17
Entering, Modifying, and Deleting Values	17
Defaults in Display Forms	18
Entries in Tables	18
PROC, REUSE, and REGEN Actions	18
Processing A Response and Highlighting	19
Special Characteristics of Display Forms	19
Window Forms	19
EXPLANATION Forms	19
Screen Protection	20
The Standard HEADER Form	20
Content	21
Working with the Form	22
Subjects in Categories	22
Segments in Data Bases	22
Related Subjects	23
Modifying Default Subject Qualifier Values	23
Processing Options	23
The Security HEADER Form	24
Security Messages	24
Chapter 3. Display Forms	27
DISPLAY FORM FIELDS	27
DISPLAY FORM ACTIONS	29
ALIASES Form	30
Accessing the ALIASES Form	30
Content	30
Working with the Form	31
Processing Options	31
CATEGORY-RELATIONSHIPS Form	32
Accessing the CATEGORY-RELATIONSHIPS Form	32
Content	32
Working with the Form	33
Processing Options	33
Specialized Actions	33
COMMAND Form	34
Content	34
Working with the Form	34
Processing Options	35
DESCRIPTION Form	35

Accessing the DESCRIPTION Form	35
Content	35
Working with the Form	35
Processing Options	36
EXPLANATION Form	37
Content	37
Working with the Form	37
Processing Options	37
Extensibility EXPLANATION Form	38
Content	38
Working with the Form	38
Processing Options	38
EXTENSIBILITY SUBJECT Form	39
Accessing the EXTENSIBILITY SUBJECT Form	39
Content	40
Working with the Form	40
Processing Options	41
Specialized Actions	41
FIELD Form	41
Accessing the FIELD Form	41
Content	41
Working with the Form	45
Processing Options	45
Specialized Actions	45
FIELD-SEGMENTS Form	46
Accessing the FIELD-SEGMENTS Form	46
Content	47
Working with the Form	47
Processing Options	48
INQUIRY ATTRIBUTES Form	48
Accessing the INQUIRY ATTRIBUTES form	48
Content	49
Working with the Form	50
Processing Options	50
JOB Form	50
Accessing the JOB Form	50
Content	51
Working with the Form	51
Processing Options	51
LCHILD Form	52
Accessing the LCHILD Form	52
Content	53
Working with the Form	53
Processing Options	53
Specialized Actions	54
LOG-SEGMENT Form	54
Accessing the LOG-SEGMENT Form	54
Contents	54
Working with the Form	55
Processing Options	56
Specialized Actions	56
LOGICAL DATABASE Form	56
Accessing the LOGICAL DATABASE Form	56
Content	56
Working with the Form	57
Processing Options	58
Specialized Actions	58
LOGICAL DATABASE STRUCTURE Form	58
Accessing the LOGICAL DATABASE STRUCTURE Form	58
Content	58
Working with the Form	59
Processing Options	60
Specialized Actions	60
MODULE Form	61
Accessing the MODULE Form	62
Content	62
Working with the Form	62
Processing Options	62
PCB Form	62
Accessing the PCB Form	62
Content	62
Working with the Form	65
Processing Options	66
PCB SENSEGS Form	66
Accessing the PCB SENSEGS Form	67

Content	67
Working with the Form	68
Processing Options	69
Specialized Actions	69
PCB SENSEG INDEXES Form	69
Accessing the PCB SENSEG INDEXES Form	70
Content	70
Working with the Form	70
Processing Options	71
PHYSICAL DATABASE Form	71
Accessing the PHYSICAL DATABASE Form	72
Content	72
Working with the Form	74
Processing Options	74
PHYSICAL DATABASE STRUCTURE FORM	74
Accessing the PHYSICAL DATABASE STRUCTURE Form	75
Content	75
Working with the Form	76
Processing Options	77
Specialized Actions	77
PHY-SEGMENT Form	78
Accessing the PHY-SEGMENT Form	78
Content	78
Working with the Form	80
Processing Options	81
Specialized Actions	81
PIP-SEGMENT Form	81
Accessing the PIP-SEGMENT Form	81
Content	82
Working with the Form	83
Processing Options	83
Specialized Actions	83
PL/I DATA Form	83
Accessing the PL/I DATA Form	84
Content	84
Working with the Form	85
Processing Options	85
PRIMARY INDEX DATABASE Form	85
Accessing the PRIMARY INDEX DATABASE Form	85
Content	85
Working with the Form	88
Processing Options	90
PROGRAM Form	90
Accessing the PROGRAM Form	90
Content	90
Working with the Form	91
Processing Options	91
PSB Form	92
Accessing the PSB Form	92
Content	92
Working with the Form	94
Processing Options	94
Specialized Actions	94
PSB-TRANSACTION Form	94
Accessing the PSB-TRANSACTION Form	94
Content	94
Working with the Form	95
Processing Options	96
RELATED ENTITIES Form	96
Accessing the RELATED ENTITIES Form	97
Content	97
Working with the Form	97
Processing Options	97
RELATIONSHIP DATA Form	98
Accessing a RELATIONSHIP DATA Form	98
Content	99
Working with the Form	99
Processing Options	100
Specialized Actions	100
SECONDARY INDEX Form	100
Accessing the SECONDARY INDEX Form	101
Content	101
Working with the Form	103
Processing Options	104
Specialized Actions	104

SECONDARY INDEX DATABASE Form	104
Accessing the SECONDARY INDEX DATABASE Form	105
Content	105
Working with the Form	107
Processing Options	107
Specialized Actions	107
No-role SEGMENT Form	107
General Comments on the SEGMENT Forms	107
Common Fields in SEGMENT Forms	108
Common Usage of SEGMENT Forms	109
Accessing the No-role SEGMENT Form	111
Content	111
Working with the Form	111
Processing Options	111
SEGMENT-FIELDS Form	112
Accessing the SEGMENT-FIELDS Form	112
Content	112
Working with the Form	114
Processing Options	115
SIP-SEGMENT Form	115
Accessing the SIP-SEGMENT Form	115
Content	115
Working with the Form	115
Processing Options	116
Specialized Actions	117
SELECTED SUBJECT-RELATIONSHIPS Form	117
Accessing the Selected SUBJECT-RELATIONSHIPS Form	118
Content	118
Working with the Form	118
Processing Options	119
Specialized Actions	119
UNSELECTED SUBJECT-RELATIONSHIPS Form	119
Accessing the Unselected SUBJECT-RELATIONSHIPS Form	119
Content	119
Working with the Form	120
Processing Options	121
Specialized Actions	121
SUBORDINATE/SUPERIOR FIELDS Form	121
Accessing the SUBORDINATE/SUPERIOR FIELDS Form	122
Content	122
Working with the Form	123
Processing Options	124
SYSDEF Form	124
Accessing the SYSDEF Form	125
Content	125
Working with the Form	125
Processing Options	125
Specialized Actions	125
SYSDEF-DATABASE Form	125
Accessing the SYSDEF-DATABASE Form	126
Content	126
Working with the Form	126
Processing Options	127
SYSDEF-PSB Form	127
Accessing the SYSDEF-PSB Form	127
Content	127
Working with the Form	128
Processing Options	129
SYSDEF-TRANSACTION FORM	129
Accessing the SYSDEF-TRANSACTION Form	129
Content	129
Working with the Form	131
Processing Options	131
SYSTEM Form	131
Accessing the SYSTEM Form	132
Content	132
Working with the Form	132
Processing Options	133
TRANSACTION Form	133
Accessing the TRANSACTION Form	133
Content	133
Working with the Form	135
Processing Options	135
USER DATA Form	135
Accessing the USER DATA Form	136

Content	136
Working with the Form	136
Processing Options	137
VLC-SEGMENT Form	138
Accessing the VLC-SEGMENT form	139
Content	139
Working with the Form	139
Processing Options	139
Specialized Actions	139
XDFLD Form	140
Accessing the XDFLD form	140
Content	140
Working with the Form	141
Processing Options	142
Chapter 4. Dictionary Batch Execution	143
Batch Operation under OS/VS	143
Operating Requirements for PLI_IN Command	144
Operating Requirements for the COBOL_IN Command	144
Operating Requirements for Sorted Glossary Output	145
Batch Operation with DL/I DOS/VS	145
Operating Requirements for the COBOL_IN Command	146
Operating Requirements for PLI_IN Command	146
Operating Requirements for Sorted Glossary Output	147
Output of Batch Dictionary Execution	147
Batch Input Streams	147
Chapter 5. Batch Forms Preparation	149
Forms Available	149
Batch Form Records	149
Mode of Input	149
Batch Form Processing	150
Access Considerations	150
Field Definition Forms	150
Header Record	152
Basic Field Attribute Record	153
Description Data Record	153
Aliases Record	154
PL/I Data Record	154
COBOL Attributes Record	155
COBOL Occurs Clause Record	156
COBOL Value Clause Record	156
Subfield Record	157
End Record	158
Field Definition Form for PL/I Data	158
Header Record	160
Basic Field Attribute Record	161
Description Data Record	161
Aliases Record	162
PL/I Data Record	162
PL/I Attributes Record	163
PL/I Picture and Initial Record	164
Subfield Record	165
End Record	166
Segment Definition Form	167
Header Record	169
Basic Segment Attributes Record	170
Description Data Record	170
Aliases Record	171
PL/I Data Record	171
Related Field Record	172
End Record	173
Text Data Form	174
Header Record	176
Input Text Record	177
End Record	178
Chapter 6. Commands	179
Command Conventions	181
The Character Set Used in Commands	182
Special Characters	182
Entering Commands	182
General Command Format	183
Keywords in Commands	184
Standard Keywords	184

Installation-defined Keywords	184
Standard Relationship Keywords	185
Installation-defined Relationship Keywords	185
Id Keywords	186
Sequence Attribute Keywords	186
Subject Name Memory (Ditto)	186
ADD Command	188
ADD_RELATIONSHIP Command	188
Restrictions on the Status of Subjects	188
Standard Relationships	189
Element-Element Relationships	189
Segment-Element Relationships	191
Data Base-Segment or PCB-Segment Relationships	192
Installation-defined Relationships	193
Adding Aliases	193
CHANGE_IN COMMAND	194
Using FROM and TO	195
CHANGE_NAME Command	196
CHANGE_RELATIONSHIP_DATA Command	197
Checkpoint Command	198
COBOL_IN Command	198
COPY Command	200
DBD_IN Command	202
DBD_OUT Command	203
DDT_OUT Command	206
DEFAULTS Command	207
DELETE Command	209
DELETE_DATA Command	209
DELETE_RELATIONSHIP Command	210
DELETE_RELATIONSHIP_DATA Command	212
DELETE_STRUCTURE Command	213
EXECUTE Command	214
EXTEND_RELATIONSHIP Command	214
The FLUSH Command	216
PLI_IN Command	217
PSB_IN Command	219
PSB_OUT Command	219
PUNCH Command	220
RECALCULATE_SEGMENT Command	221
RELOCATE Command	222
REPORT Command	224
Subject-Specific Reports	224
Display Form Equivalent Reports	227
Indirect Subject Reference Reports	227
Glossary Reports	229
Guide Reports	231
SCAN Command	232
Character Strings	236
The SET Commands	236
SIGN_ON Command	238
STAGE_1_OUT Command	239
STRUCTURES_OUT Command	240
Appendix A: Dictionary Keywords for Subject Data	243
Database	244
Physical Attributes of the Data Base	244
Physical Characteristics of Data Sets	246
Free-Form Description	249
User Data	249
Element	250
General Attributes	250
COBOL Attributes	252
COBOL VALUE Clause	253
COBOL OCCURS Clause	253
PL/I Attributes	254
Inquiry header and edit description	256
Language Commentary and PL/I Data	258
Free-Form Description	259
User Data	259
PCB	260
Attributes of a Data Base PCB	260
Attributes of an Output Message PCB	261
Free-Form Description	262
User Data	262
Segment	263

General Attributes	263
Language Commentary and PL/I Data	264
Fields Used for Secondary Index	265
SRCH Fields for Generating Key Field of Secondary Index	
Data Base Root Segment	266
SUBSEQ and DDATA Fields from Index Source Segment	267
Free-Form Description	268
User Data	268
System	269
Attributes of the Program	269
Attributes of the PSB	270
Attributes of the Transaction	272
Free-Form Description	274
User Data	274
Appendix B: Dictionary Keywords for Relationship Data	275
DATABASE with SEGMENT	276
General Characteristics	276
Physical Attributes of Segment	278
Physical Child SEGMENT	280
Logical Child Segment	281
Logical Parent Segment	283
Source Segment (Virtual or Logical Segment)	284
PCB with Segment	286
General Characteristics	286
Index Used in SSAs	287
Segment with Element	288
General Characteristics	288
Element Contains Element	289
General Characteristics	289
SYSDEF WITH DATABASE	290
General Characteristics	290
SYSDEF WITH PSB	291
General Characteristics	291
SYSDEF WITH TRANSACTION	292
General Characteristics	292
PSB WITH TRANSACTION	293
General Characteristics	293
Dictionary Messages and Codes	295
Dictionary Abnormal Termination Codes	295
Dictionary Numbered Messages	296
General System Messages (DBD00nn t)	296
Language Translator (DBD01nn t)	301
DBD_IN (DBD02nn t)	308
DBD_OUT (DBD03nn t)	309
PSB_IN (DBD04nn t)	310
PSB_OUT (DBD05nn t)	311
REPORT (DBD06nn t)	312
STRUCTURES_OUT (DBD07nn t)	314
SCAN (DBD08nn t)	321
COPY (DBD09nn t)	322
ADD (DBD10nn t)	324
CHANGE_IN (DBD11nn t)	324
DELETE (DBD12nn t)	324
CHANGE_RELATIONSHIP_DATA (DBD13nn t)	324
ADD_RELATIONSHIP (DBD14nn t)	325
CHANGE_NAME (DBD16nn t)	326
ADD_RELATIONSHIP with data (DBD17nn t)	326
ADD with data (DBD18nn t)	326
DELETE_DATA (DBD19nn t)	326
DELETE_RELATIONSHIP_DATA (DBD20nn t)	326
COBOL_IN and PLI_IN (DBD21nn and DBD22nn t)	327
General Service /Common Function (DBD23nn t)	334
Interactive Display Forms Facility (DBD24nn t to DBD40nn t)	334
DELETE_STRUCTURE (DBD41nn t)	349
RELOCATE (DBD42nn t)	349
GIS Support (DBD43nn t)	351
Batch Input Forms (DBD44nn t)	351
EXTEND_RELATIONSHIP (DBD45nn t)	357
PUNCH (DBD49nn t)	359
IMS/VS System Definition Support (DBD50nn t)	360
Default (DBD51nn t)	361
Checkpoint (DBD52nn t)	361
Program Access (DBD53nn t)	361

CICS Support (DBD54nn t)	362
Install (DBD55nn t)	366
General Extensibility Display Forms (DBD56nn t)	373
CATEGORY-RELTYPES or RELATIONSHIPS Messages (5640-49)	376
Sign_On (DBD60nn t)	378
Index	385

FIGURES

Figure 1.	Typical Information in the Definitions for Different Subject Categories	3
Figure 2.	Table of Subject Name Rules	6
Figure 3.	Display Forms and Their Possible Sequences	14
Figure 4.	Example of Display Form and Alternative Action Areas	17
Figure 5.	HEADER Form	20
Figure 6.	Initial Security HEADER Form	25
Figure 7.	Security HEADER Form After Sign-On	26
Figure 8.	Subject Name Rules for Display Forms	28
Figure 9.	ALIASES Form	31
Figure 10.	CATEGORY-RELATIONSHIPS Form	32
Figure 11.	DB/DC Dictionary COMMAND Form	34
Figure 12.	DESCRIPTION Form	36
Figure 13.	EXPLANATION Form	37
Figure 14.	Extensibility EXPLANATION Form	38
Figure 15.	EXTENSIBILITY SUBJECT Form	39
Figure 16.	FIELD Form	42
Figure 17.	FIELD-SEGMENTS Form	46
Figure 18.	INQUIRY ATTRIBUTES Form	48
Figure 19.	JOB Form	51
Figure 20.	LCHILD Form	52
Figure 21.	LOG-SEGMENT Form	55
Figure 22.	LOGICAL DATABASE Form	57
Figure 23.	LOGICAL DATABASE STRUCTURE Form	59
Figure 24.	MODULE Form	61
Figure 25.	PCB Form	63
Figure 26.	PCB SENSEGS Form	66
Figure 27.	PCB SENSEG INDEXES Form	70
Figure 28.	PHYSICAL DATABASE Form	71
Figure 29.	PHYSICAL DATABASE STRUCTURE Form	75
Figure 30.	PHY-SEGMENT Form	78
Figure 31.	PIP-SEGMENT Form	82
Figure 32.	PL/I DATA Form	84
Figure 33.	PRIMARY INDEX DATABASE Form	86
Figure 34.	PROGRAM Form	91
Figure 35.	PSB Form	92
Figure 36.	PSB-TRANSACTION Form	95
Figure 37.	RELATED ENTITIES Form	96
Figure 38.	RELATIONSHIP DATA Form	98
Figure 39.	SECONDARY INDEX Form	101
Figure 40.	SECONDARY INDEX DATABASE Form	105
Figure 41.	No-role SEGMENT Form	110
Figure 42.	SEGMENT-FIELDS Form	113
Figure 43.	SIP-SEGMENT Form	116
Figure 44.	Selected SUBJECT-RELATIONSHIPS Form	117
Figure 45.	Unselected SUBJECT-RELATIONSHIPS Form	120
Figure 46.	SUBORDINATE/SUPERIOR FIELDS Form	122
Figure 47.	SYSDEF Form	124
Figure 48.	SYSDEF-DATABASE Form	126
Figure 49.	SYSDEF-PSB Form	128
Figure 50.	SYSDEF-TRANSACTION Form	130
Figure 51.	SYSTEM Form	132
Figure 52.	TRANSACTION Form	134
Figure 53.	USER DATA Form	136
Figure 54.	Alternative USER DATA Form	137
Figure 55.	VLC-SEGMENT Form	138
Figure 56.	XDFLD Form	140
Figure 57.	OS/V5 JCL for Batch Dictionary Operation	143
Figure 58.	DOS/V5 JCL for Batch Dictionary Execution	145
Figure 59.	Typical Structure of Dictionary Batch Input	148
Figure 60.	Data Dictionary Field Definition Form	151
Figure 61.	Data Dictionary Field Definition Form for PL/I Data	159
Figure 62.	Data Dictionary Segment Form	168
Figure 63.	Data Dictionary Text Data Form	175
Figure 64.	General Command Format	183
Figure 65.	Examples of Command Syntax	187
Figure 66.	Effect of GSAMDBD Operand in DBD_OUT Command on	

	ACCESS=Value	205
Figure 67.	Response to the DEFAULTS Command	208
Figure 68.	Valid DETAIL parameters	226
Figure 69.	Valid Report Names for NAME=reportname	228

CHAPTER 1. DICTIONARY SUBJECTS AND SUBJECT CATEGORIES

The Dictionary can contain four types of data: subject names, subject data, relationships, and relationship data. This chapter describes the four types of data and the rules that govern their formation and relationships. It also discusses Dictionary installation options that affect the way data enters the Dictionary.

SUBJECT CATEGORIES AND SUBJECT NAMES

The Dictionary allows you to define subjects in several "standard" subject categories. Three of the subject categories are for data bases or data sets and their component parts. The three categories are:

DATABASE (or FILE)	for physical, index, or logical DL/I data bases or non-DL/I data sets
SEGMENT (or RECORD)	for DL/I data base segments or non-DL/I records
ELEMENT (or FIELD)	for data fields or group data items

The other eight subject categories are intended for definitions of components of application systems that use the data:

JOB	for processing units corresponding to a single job
MODULE	for functional subsections of a program
PCB	for DL/I Program Communication Blocks (PCBs)
PROGRAM	for major jobs or job steps
PSB	for DL/I Program Specification Blocks (PSBs)
SYSDEF	for IMS/VS system definitions
SYSTEM	for major application systems, such as a payroll system
TRANSACTION	for IMS or non-IMS transactions

In addition to the eleven standard categories mentioned above, your installation may define subjects in other "installation-defined" categories.

Other categories exist in the context of control of the content of the Dictionary. They are DDUSER, which is concerned with Dictionary security, and the Extensibility Control Information (ECI) categories, ATTRTYPE, RELTYPE, and CATEGORY. These categories are fully discussed in the DB/DC Data Dictionary Administration and Customization Guide.

There are several types of entries you can make as part of a definition in the Dictionary:

Subject Name

The subject name is composed of a user name for the subject, and subject name qualifiers that make the name unique to the Dictionary. See "Subject Name and Subject Name Rules" below for more information.

Aliases

Aliases are other names by which a subject may be referenced. The primary name and its aliases all have the same meaning (that is, an alias cannot have any attributes or relationships of its own).

Subject Data

Subject data is any data associated with a subject that is independent of the subject's relationships with other subjects.

attributes

anticipated entries for various characteristics of subjects in a given subject category.

description

text describing the subject, with no required format or content except for line length and line numbering.

User Data

any information the user wants to put in the Dictionary that is not included in the anticipated attributes or relationships, with no restrictions except for line length and line numbering. Each subject may have up to five sets of User Data associated with it.

language commentary or PL/I data

entries of text to be handled as language commentary or additional PL/I attributes, for segments and elements (no restrictions except line length and line numbering).

Relationships

Relationships are cross-references between a given subject and other subjects in the same or other categories.

Relationship Data

Relationship data is any information about the nature of the relationship between two subjects, for example the starting position of an element in a segment.

DEFINITIONS IN SUBJECT CATEGORIES

Not all the types of Dictionary entries appear in all subject categories. Figure 1 summarizes some of the entries that may be made for the different subject categories. In Dictionary commands, entries for attributes, relationships, and relationship data are represented by keywords. You can find attribute keywords in Appendix A and relationship and relationship data keywords in Appendix B. (The standard categories SYSDEF, SYSTEM, JOB, PROGRAM, MODULE, TRANSACTION, and PSB are grouped together as system subjects in the figure.)

Figure 1. Typical Information in the Definitions for Different Subject Categories

	SYSTEM CATEGORIES					
<p>DATABASE (data set)</p> <p>Name</p> <p>Attributes Block sizes Access methods Names of data definition statements Name of user-supplied randomizer Number of direct access anchor points Indicators for data set groups</p> <p>Description</p> <p>User Data</p> <p>Relationships</p> <p>Relationship Data Parent-child relationships Hierarchical structure of segments Rules for insert, delete, and replace</p>	<p>SEGMENT (record)</p> <p>Name</p> <p>Attributes Length (fixed or variable) PL/I alignment Effective date</p> <p>Description</p> <p>User Data</p> <p>PL/I Data</p> <p>Relationships</p> <p>Relationship Data Level</p>	<p>ELEMENT (field)</p> <p>Name</p> <p>Attributes Length Type COBOL attributes PL/I attributes Inquiry attributes Effective date</p> <p>Description</p> <p>User Data</p> <p>PL/I Data</p> <p>Relationships</p> <p>Relationship Data Start position within segment Contained in another field</p>	<p>PCB</p> <p>Name</p> <p>Attributes Key length Type</p> <p>Description</p> <p>User Data</p> <p>Relationships</p> <p>Relationship Data Sensitive segment structure Processing options</p>	<p>SYSTEM Name Description User Data Relationships</p> <p>JOB Name Description User Data Relationships</p> <p>PROGRAM Name Attributes Language Type Size Description User Data Relationships</p> <p>MODULE Name Description User Data Relationships</p>	<p>TRANSACTION Name Attributes Type Code Description User Data Relationships</p> <p>PSB Name Attributes Language DL/I parameters Description User Data Relationships</p> <p>SYSDEF Name Description User Data Relationships Relationship Data Type of Scheduling</p>	<p>INSTALLATION DEFINED CATEGORIES</p> <p>Name</p> <p>Attributes</p> <p>Description</p> <p>User Data</p> <p>Relationships</p> <p>Relationship Data</p>

Every entry you put in the Dictionary is associated with a specific category or (for a relationship entry) a pair of subject categories. Every entry must also be associated with a subject name or (for relationships) a pair of subject names because a subject name is required for each definition stored in the Dictionary.

SUBJECT NAMES AND SUBJECT NAME RULES

"Subject name" in the Dictionary has a quite precise meaning. A subject name has four parts: the user name for the subject, plus three subject name qualifiers. Subject names are always stored in the Dictionary with all three qualifiers. The fully qualified subject name, when used in a command, has the following format:

(status,subjectcode,username,occurrence)

Note: When the name is stored in the Dictionary, no parentheses or commas are stored.

The user name is the name you give the subject. The name must be no more than 31 alphameric characters long. In addition, for controlled output for the DBD_OUT, PSB_OUT, DDT_OUT STRUCTURES_OUT, and STAGE_1_OUT commands, the names should conform to the rules for the programming language or system with which they are associated.

The first character of the user name must be an alphabetic or numeric character, or one of the following: \$ (dollar sign), # (pound sign), @ (at), . (period), or / (slash). The remaining characters may also include _ (underscore), or - (hyphen).

The first subject name qualifier is the status qualifier. This is a single-character code that designates the status of the subject in your system. The alphabetic characters A through T (except P), and the numeric characters 0 through 9, are all available to be assigned by your installation. These codes are generally used to denote various levels of test status of a Dictionary subject. The status code allows you to differentiate between several versions of a Dictionary definition. Different versions can be assigned different status codes, but retain the same user name, subject code, and occurrence number. The Dictionary reserves the status code of P to have a special meaning of production, that is, currently in use in your system.

The second subject name qualifier is the subject code. It is a single-letter code with different meanings for different categories. For the ELEMENT and SEGMENT categories, the subject code designates the programming language associated with the subject name, and identifies the subject as belonging to a DL/I data base or to a non-DL/I data set. For the DATABASE category, the subject code identifies the type of data base or data set. For the remaining standard categories, the subject code is simply a letter code for that category (for example, T for transaction). Installation-defined categories do not have a subject code qualifier associated with them.

The final subject name qualifier (appearing last in the subject name, after the user name) is occurrence. This designates the subject name as one of multiple occurrences of an otherwise identical subject name. For most categories, the occurrence number can vary from 0 to 255. It does not imply anything about the similarity of characteristics of the subjects that are other occurrences of the subject name. The occurrence qualifier is useful for installations that have not yet eliminated duplicate names for programs or that have many different elements or segments and the same user name.

For PCBs, the occurrence number has a special meaning. It gives the sequential position of a PCB within the PSB to which it belongs. The remaining portions of the subject names for a PSB and the PCBs defined within it are usually, but not necessarily, identical.

The possible values for the four parts of the subject name vary according to the subject category. Figure 2 summarizes the qualifier values for subject names in different categories. The notes that follow the figure explain the rules for forming user names and assigning subject name qualifiers.

SUBJECT CATEGORY (ALTERNATIVE FORMS)	SUBJECT NAME			
	STATUS (See Notes 1&2)	SUBJECT CODE	USER NAME (See Note 4)	OCCURRENCE NUMBER
DATABASE (D, DBS, FILE, L, FIL)	A-T or 0-9. All are test levels except P which is Production.	P physical DL/I L logical, DL/I F file (data set), non-DL/I X primary index data base Y secondary index data base	1-31 characters. Should follow DL/I or non-DLI systems naming conventions	0 for DL/I. 0 to 255 for non-DL/I
SEGMENT (S, SEG, RECORD, R, REC) ELEMENT (E, DTE, FIELD, FLD)		A assembler language DL/I B PL/I, DL/I C COBOL, DL/I J assembler language, non-DL/I K PL/I, non-DL/I L COBOL, non-DL/I	1-31 characters. Should follow rules of programming language	0 to 255
PCB (P)		P PCB (See Note 3)	1-31 characters. Usually same as corresponding PSB	1 to 255 (sequence of PCBs in PSB). (See Note 3)
SYSDEF (N, SDF)		N	1-31 characters.	0 to 255
SYSTEM (Y, SYS)		S	1-31 characters.	0 to 255
JOB (J)		J	1-31 characters. Should follow operating system naming convention	0 to 255
PROGRAM (G, PGM)		G		0 to 255
MODULE (M, MOD)		M		0 to 255
TRANSACTION (T, TRN, TRANSACT)		T		0 to 255
PSB (B)		P		1-31 characters. Should follow DL/I naming conventions
Installation-defined	none	1-31 characters	0 to 255	

Figure 2. Table of Subject Name Rules

Notes to Table of Subject Name Rules

1. The Dictionary places some restrictions on the status that can be assigned to related subjects in a Dictionary hierarchical structure. The status rules for all categories are:
 - No status but P under P.
 - P status may occur under any status.
 - A subject with test status (A-T except P, or 0-9) can have only subjects with the same test status or P status under it.
 - An alias must have the same status as the primary name to which it belongs.

From these rules, it follows that there can be no more than two status types in one Dictionary structure; if there are two, one must be P.

Chapter 5 of the DB/DC Data Dictionary Applications Guide has further discussion on Dictionary structures; Chapter 12 gives the hierarchy rules for installation-defined subject categories.

2. Your installation may have included, as part of its Dictionary installation, entries defining the structure of the Dictionary data bases themselves with an assigned status code of D. To avoid possible conflict with user-entered definitions, the D status should be reserved.
3. PCBs do not have names in DL/I, but in the Dictionary they usually have the same user name as the PSB to which they belong, and an occurrence number that gives the sequential position of the PCB within the PSB.

The subject code P is required in input of PCB names but does not appear in output. For example, a PCB with a subject name entered as (P,P,PSBONE,10) will appear as P PSBONE 10 in a report.

For PCB names, you can enter a four-part subject name or a user name with occurrence number. See "Subject Name Qualifier Defaults" below. The Dictionary allows occurrence numbers 1 to 255 for PCBs; data base PCBs, however, should have an occurrence number no greater than 32.

4. For all subject categories, the Dictionary accepts any name up to 31 characters long. But to ensure you have usable names, follow the additional restrictions recommended in this figure.

You do not have to follow these recommendations for all the aliases of a given subject name. Just be sure the subject has one name, either primary or alias, that is suitable for output. For example, if you define an element that may be used in assembler language, PL/I, or COBOL programs in your DL/I system, you may wish to assign to it at least three names, with subject codes A, B, and C, each conforming to the designated language conventions. It is possible, however, to produce structures for all languages using the same name; if the primary name is an assembler language name (subject code A), it is acceptable for all three languages.

For some categories, names over eight characters cannot be accessed or entered on display forms. See Figure 2 for more information.

SUBJECT NAME QUALIFIER DEFAULTS

The Dictionary always stores a complete subject name, including the status, code, user name, and occurrence number. It is possible, however, for you to designate subjects in commands by their user names alone. If you enter the user name without qualifiers in a command, the Dictionary adds default subject name qualifiers. These default values are defined by your installation when the Dictionary is installed.

STAT Specifies the default status.

LANG Specifies the default language code for subjects that are segments or elements.

DBTP Specifies the default data base code for data base subjects.

OCR Specifies the default occurrence number of the subject name.

MODIFYING INSTALLATION-DEFINED DEFAULT VALUES

Whether you are an online or a batch user you may modify the subject name qualifier default values. Online users can modify the values on the HEADER display form. The batch user (or the interactive user using the COMMAND form) may modify the default values by using the SETSTAT, SETLANG, SETDBTP, or SETOCR commands described in Chapter 6.

In addition to the subject name qualifier default values above, there are default values for other Dictionary functions. Chapter 6 contains descriptions of these functions and the commands that control them (SETCKPT, SETLOGU, SETLOGI, and FLUSH).

The installation-defined default values for all system defaults can be obtained by using the DEFAULTS command.

SECURITY CONSIDERATIONS

If your installation has chosen to operate the Dictionary with security enabled, your access to the Dictionary may be restricted. The installation controls the data you may access and the modifications or additions you may make with an access profile. The data base administrator (DBA) defines an access profile for each user. The profile specifies whether you may use the COMMAND form. It also contains a list of all the categories you may access. Two more lists contain status codes. One specifies the status codes you may view, the other the codes you may update. To be able to modify a given subject, your access profile must contain the category of the subject and the status code from the subject name.

CHAPTER 2. USING THE DICTIONARY ONLINE

USING THE DICTIONARY ONLINE WITH IMS/VVS

When you use the Dictionary online with IMS/VVS you may enter either Dictionary commands or definitions on display forms. You can use the commands either on the COMMAND Form or in an online mode (CMNDMOD, DBAMOD, IOMOD, OR UPDTMOD). Chapter 3 contains a section on the Command form. The online modes are not discussed in this book. See the DB/DC Data Dictionary Administration and Customization Guide for more information about them.

If you are to use the Dictionary online, the operator should use the JCL that is normally used to bring up the IMS/VVS DB/DC control region. Then the operator should start a message region in the normal way for your installation (for example, by executing the IMS/VVS procedure IMSMSG). The Dictionary runs as a conversational program in the message region. For more information about bringing up the Dictionary online, refer to Chapter 2 of the DB/DC Data Dictionary Administration and Customization Guide.

STARTING A SESSION

To specify that you wish to use the display station to enter Dictionary commands or to enter, review, or edit information on display forms, key in:

/EXIT

/EXIT clears the scratch pad area (SPA) to avoid any possibility of another user's values being carried over into your session. When /EXIT completes, IMS/VVS responds:

DFS058 EXIT COMMAND COMPLETED

Then key in:

/FORMAT value

The valid values for /FORMAT are HDRMOD, HDRSEC, DISPMOD, and DISPSEC. Entering /FORMAT with one of these values results in the display of the HEADER form. The HEADER form is described in detail later in this chapter.

If Dictionary security is not enabled, use the HDRMOD and DISPMOD values. The command /FORMAT HDRMOD results in the display of the standard HEADER form that allows you to use display forms and Dictionary commands during the same session. If you use this command you will be able to access display forms in edit, delete, or display mode. The command /FORMAT DISPMOD results in the display of the standard HEADER form that allows you to access other display forms in display mode only. Display mode allows you to review the information on the forms, but you may not modify or delete the information.

If Dictionary security is enabled, you will have to use the HDRSEC or DISPSEC values to access the display forms. The command /FORMAT HDRSEC results in the display of a security HEADER form that may require you to enter a sign-on and a password before you may access the display forms. If you use this command you will be able to access display forms in edit, delete, or display mode. The command /FORMAT DISPSEC results in the display of a security HEADER form that, after sign-on, allows use of the display forms in display mode only.

TERMINATING A SESSION

You can terminate a session by selecting the QUIT action on the HEADER form. (You may return to the HEADER form from any other form by selecting the HDR action.)

OUTPUT FROM IMS/VS ONLINE OPERATIONS

Reports and messages are normally routed to the initiating display station; punch output is normally routed to the SPOOL2 punch. You can, however, change these routings by specifying the destination parameter (DEST) when using some commands on the COMMAND form. A copy of all input commands, internally generated commands, and messages may be routed to the SPOOL1 printer. You will have to ask the computer operator to schedule the Spool Print program if you do not receive your output.

If punched output is specified to a SYSOUT=B data set, you will have to ask the computer operator to start a class B writer when the IMS/VS control region is terminated.

In the interactive display forms facility, one line is reserved at the bottom of each display form for Dictionary or IMS/VS system messages.

USING THE DICTIONARY ONLINE WITH CICS/VS

Using the Dictionary online with CICS/VS gives you access to the Dictionary through the interactive display forms facility. Two regions (partitions in DOS/VS and OS/VS1) are required: the CICS/VS region, and a region for the Dictionary. An application program in the CICS/VS region allows communication between the two regions when you wish to use the Dictionary.

To use the Dictionary, it is necessary for the operator to start the Dictionary's region.

To enable communication with the Dictionary, enter the transaction:

DDSU

The transaction identifier (DDSU) is an example. This identifier may be defined by your installation. Make sure you use the correct identifier for your installation. You may enter the transaction before starting the batch region, but CICS/VS must first be active.

When you enter the transaction, you will receive the response:

DBD5410 DICTIONARY ENVIRONMENT START-UP SUCCESSFUL

If another user entered the transaction before you, you will receive the response:

DBD5417 ENVIRONMENT ALREADY INITIALIZED

Either of these messages means you may attempt to start a session with the Dictionary.

STARTING A SESSION

To start a session with the Dictionary, enter the transaction:

DICT MOD=modname

The transaction identifier (DICT) may be installation defined, so make sure you use the identifier that is valid at your installation. The valid values for modname are HDRMOD, HDRSEC, DISPMOD, and DISPSEC. Entering the transaction above with one of

these values for modname displays a standard or security HEADER form in edit or display mode, depending upon which modname value you specify.

If Dictionary security is not enabled, use the HDRMOD and DISPMOD values. Selecting the HDRMOD value results in the display of the standard HEADER form that allows you to use display forms and Dictionary commands during the same session. If you use this value in the transaction, you will be able to access display forms in edit, delete, or display mode. Selecting the value DISPMOD results in the display of the standard HEADER form that allows you to access other display forms in display mode only. Display mode allows you to review the information on the forms, but you may not modify or delete the information.

If Dictionary security is enabled, you will have to use the HDRSEC or DISPSEC values to access the display forms. Selecting the value HDRSEC results in the display of a security HEADER form that may require you to enter a sign-on and a password before you may access the display forms. If you use this value you will be able to access display forms in edit, delete, or display mode. Selecting the value DISPSEC results in the display of a security HEADER form that, after sign-on, allows use of the display forms in display mode only.

TERMINATING A SESSION

You can terminate a session by selecting the QUIT action on the HEADER form. (You may return to the HEADER form from any other form by selecting the HDR action.)

SHUTTING DOWN THE DICTIONARY ENVIRONMENT

If you have the authorization, you can shut down the Dictionary environment. You can cause the shutdown to be normal or immediate.

You can shut down the Dictionary normally by entering the transaction:

```
DDSD MODE=NORMAL
```

The transaction identifier (DDSD) is an example. This identifier may be defined by the installation. Make sure you use the correct identifier for your installation.

After all active sessions have terminated, the Dictionary itself will terminate. No new sessions may begin.

If an immediate shutdown of the Dictionary is required, enter the transaction:

```
DDSD MODE=IMM
```

Specifying DDSD MODE=IMM ends each session immediately when an input action is selected but allows the output from the input action previous to the shutdown request to be produced.

If an emergency shutdown of CICS/VS is required, prior to requesting the emergency shutdown enter the transaction:

```
DDSD MODE=FORCE
```

A forced shutdown of the Dictionary causes the termination of all Dictionary processing without regard to the status of the sessions with the Dictionary.

The transaction identifier (DDSD in the examples) terminates all the sessions with the Dictionary, and terminates the Dictionary.

You may want to know how many sessions are active; if so, enter the transaction:

DDSD MODE=INQUIRE

You may change the shutdown mode from NORMAL to IMM, NORMAL to FORCE, or IMM to FORCE by using the DDSD MODE= transaction and specifying the mode you wish to change to.

OUTPUT FROM CICS/VS ONLINE COMMANDS

Segmented Output

When entering commands under CICS/VS, some responses to your commands may be segmented because of insufficient storage; if the responses are segmented, the screen clears and the following message is displayed at the top of the screen:

```
DBD5430 I NO STORAGE, OUTPUT SEGMENTED,  
PRESS ENTER TO CONTINUE
```

After you press ENTER, the original COMMAND form is redisplayed with your command and the first frame of output. You can page through the first segment of output, by pressing the PA1 key until the screen clears again and displays the following message at the top of the screen:

```
DBD5431 I END OF SEGMENT, YES TO REPEAT,  
NO TO DISPLAY NEXT SEGMENT
```

At this point you can return to the beginning of the display segment you are viewing or look at the next segment.

It is imperative that you are aware of the fact that when you are running the Dictionary under CICS/VS and paging through your segmented output, no other terminal users may access the Dictionary. Until you have processed your segmented output, NO OTHER USER MAY PROCESS A DICTIONARY REQUEST!

Routing of Output

Reports and messages are normally routed to the initiating display station; punch output is normally routed to the DDPUNCH (SYSPCH for DOS) punch. All internally generated commands and messages are routed to the DDLIST (SYSLSLST for DOS) printer.

If punched output is specified to a SYSOUT=B data set, you may have to tell the computer operator to start a class B writer.

USING DISPLAY FORMS

If your Dictionary installation has made the interactive display forms facility available, you will probably find the display forms a convenient way to do most of your Dictionary work. This facility displays standard display forms on whatever terminal your installation has designated. The display forms are a set of predefined display images, each with a fixed format of labeled fields and columns that provide all the information you need for a specific definitional task.

You can use the display forms to review definitions already stored in the Dictionary (no matter how they were entered), to enter new definitions, or to modify or delete stored definitions.

The display forms are designed for your convenience. They make it unnecessary for you to memorize the command syntax. They guide your entries to help you avoid omitting parameters, specifying contradictory information, or making spelling errors in entering keywords.

The rest of this introduction gives you general information about the use of the display forms. The descriptions of the individual forms that follow in Chapter 3 assume that you are familiar with this general information, as well as with the general operation of the terminal selected for use with the Dictionary.

KINDS OF DISPLAY FORMS

Figure 3 shows the available display forms and their possible sequences. The first is a HEADER form, which is central to your use of the display forms. It allows you to access any one of the first level or key forms that lead to all other forms in the facility. Because of its importance to the process of using display forms, a section describing the HEADER form's characteristics and use is included near the end of this chapter. The remainder of the forms are described in Chapter 3.

The COMMAND form also differs from all the other forms. It allows you to enter Dictionary commands directly and receive the responses they generate. For more information about the COMMAND form see Chapter 3. The commands themselves are explained in Chapter 6.

Dictionary commands allow you to make any kind of change or addition to the Dictionary data bases. In addition to being usable on the COMMAND form, they may be used in the online modes (CMNDMOD, DBAMOD, IOMOD, and UPDTMOD), under IMS/VS and in batch operation. The online modes are not discussed in this book; further information may be found in the DB/DC Data Dictionary Administration and Customization Guide.

The key forms (shown on the first row below the HEADER) in general correspond to Dictionary categories. The DATABASE category is divided between the PHYSICAL, LOGICAL, PRIMARY INDEX, and SECONDARY INDEX DATABASE forms. The SEGMENT category is similarly subdivided into the no-role SEGMENT, PHY-SEGMENT, LOG-SEGMENT, PIP-SEGMENT, SIP-SEGMENT, and VLC-SEGMENT forms. All other categories, FIELD, PCB, SYSTEM, SYSDEF, JOB, PROGRAM, MODULE, TRANSACTION, and PSB are represented by key forms of the same names. The EXTENSIBILITY SUBJECT form is the key form used to edit and display definitions of subjects in installation-defined categories. The final key form, the RELATED ENTITIES form, is for entry of relationships that, in most cases, do not have any relationship data associated with them.

The subordinate forms (shown below the first full row) accept information that supplements the definitions on the key forms, from which they can be requested. The common subordinate forms grouped at the bottom of the figure can each be requested from more than one form. The numbers below each form show the common subordinate forms you can request from each.

Some forms also have a W below them. The forms so marked are windows, or partial displays of a stored list that may be much larger than the viewing surface that the terminal can present at one viewing. These window forms allow you to move up and down over the stored list.

Figure 3 (Part 1 of 2). Display Forms and Their Possible Sequences

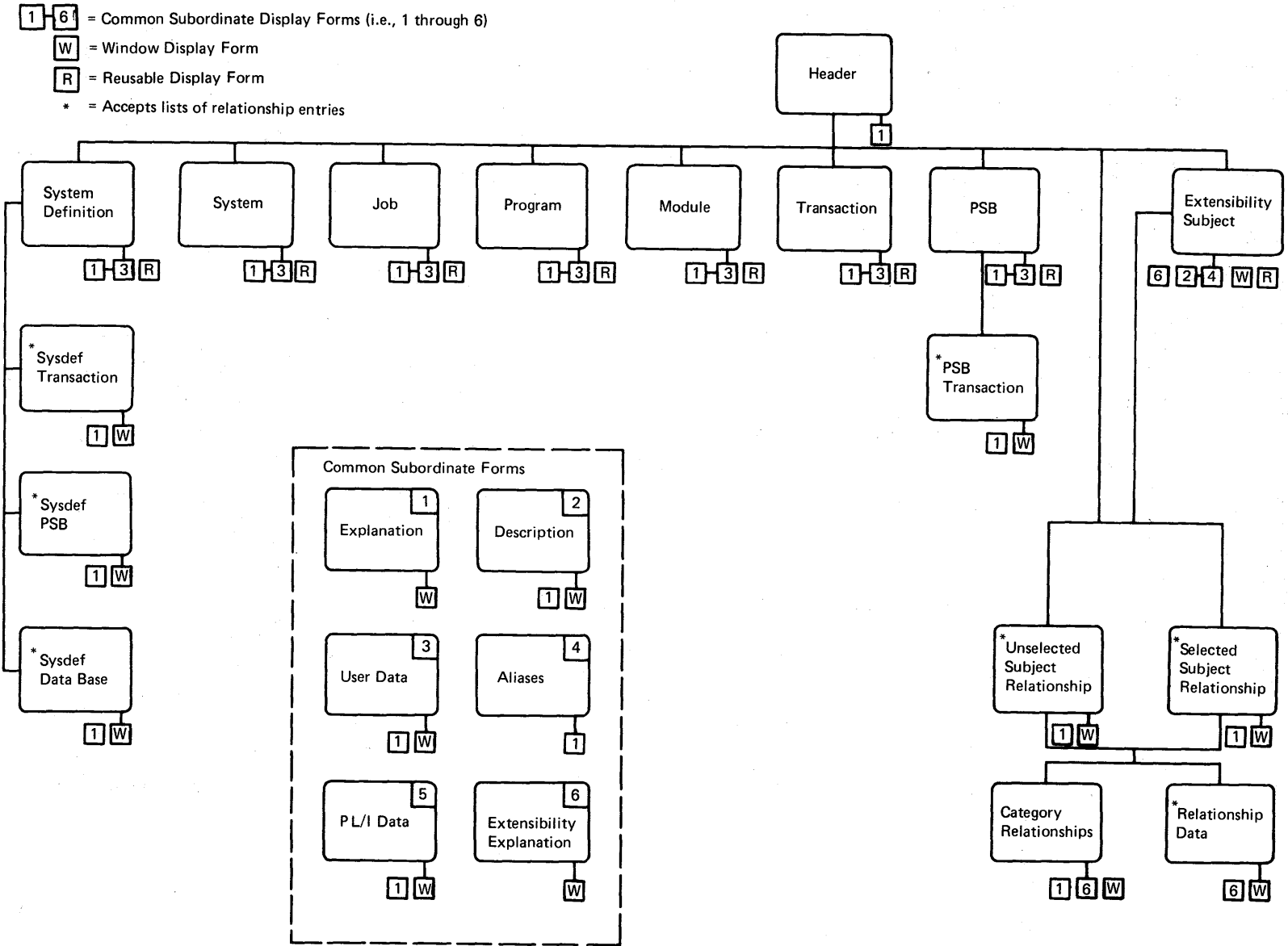
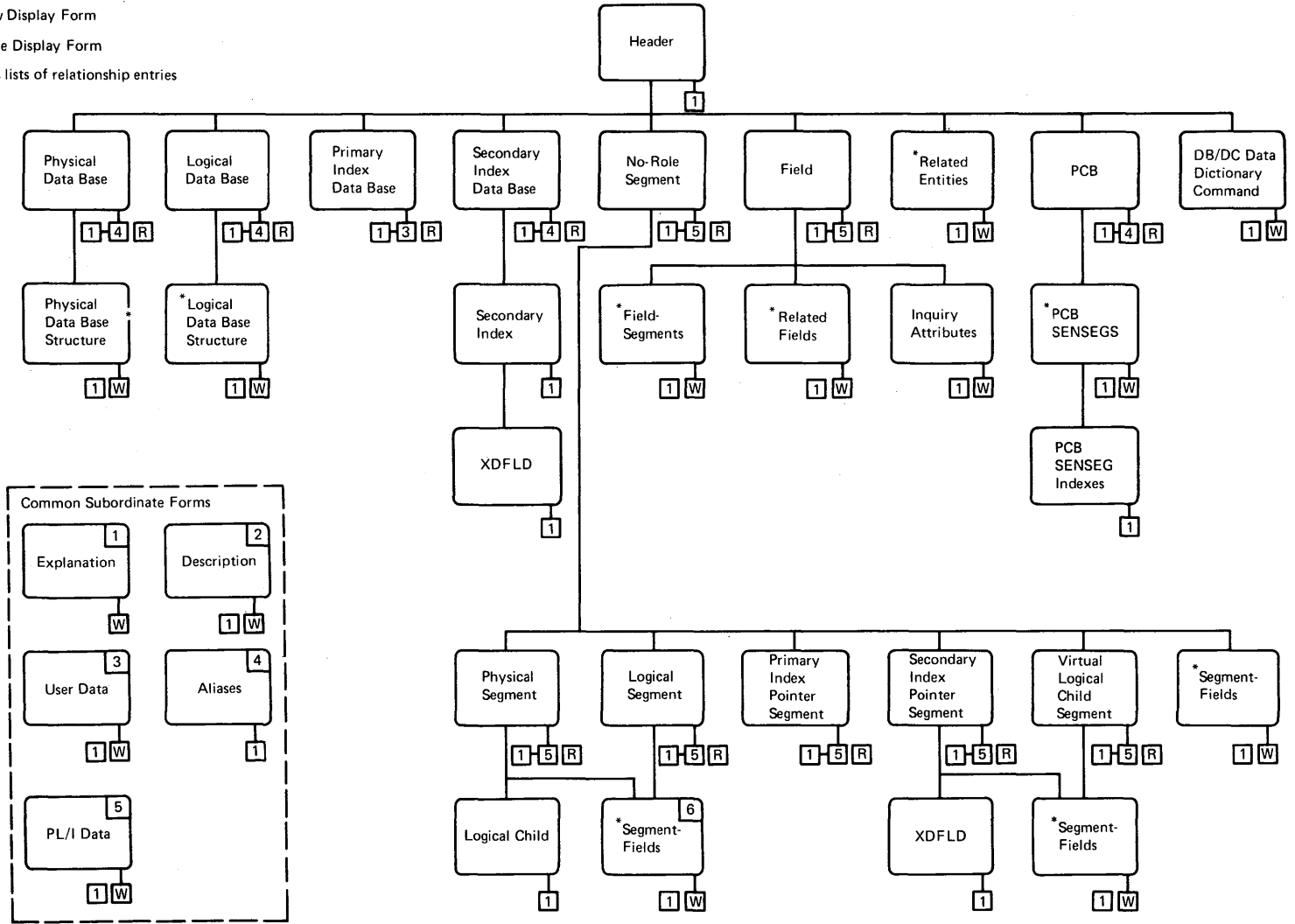


Figure 3 (Part 2 of 2). Display Forms and Their Possible Sequences

- 1-5 = Common Subordinate Display Forms (i.e., 1 through 5)
- W = Window Display Form
- R = Reusable Display Form
- * = Accepts lists of relationship entries



Relationship entries may be added to the Dictionary via the RELATED ENTITIES form and special forms for relationship data. The display forms that accept lists of relationship entries are marked with an asterisk next to their title in the figure.

GENERAL FORMAT AND PROCEDURES

Each display form is identified by a title in the upper left corner, as shown in the example in Figure 4. For most forms this title is unique, but for certain multipurpose forms, such as the SUBORDINATE/SUPERIOR FIELDS form, a part of the title changes according to the type of data displayed.

Except for the HEADER form, most of each form consists of fields and tables in which you can enter parameters for the kind of subject defined on the form.

The bottom of each form is an area for processing options, set off from the rest of the form by a double line. The processing options represent actions that allow you to control processing of the information entered, to request additional information about the use of the form, or to request display of another form.

Each action is preceded by a number that identifies the action. Some actions have fields associated with them that must be filled in before the action is selected. You can select an action by using any of the following techniques:

- Enter the number of the desired action in the field labeled ACT, which always appears in the upper left corner of the action area; then press the ENTER key.
- Press the program function (PF) key numbered the same as the action you want.
- Press the ENTER key on forms with the PROC action to default to that action.

The bottom line on all forms is a response line. This line is used to display messages about the results obtained when processing data you have entered on the form. The same line is used to display messages from IMS/VS.

The display forms make use of the highlighting feature of the terminal. That is, some parts of the image are displayed with a higher intensity (in highlighting) to give them more emphasis than the lower intensity background. In general, anything in highlight represents data stored in the Dictionary. All the preprinted field and column labels and any entries just made and not yet processed are not highlighted.

Modes of Operation

The HEADER form is used to indicate the mode of operation you want. Three modes are available: edit, delete, and display. Edit mode allows you to add data to the Dictionary by filling in forms and processing them. You may also alter information that has already been stored in the Dictionary. Delete mode allows you to delete information from the Dictionary. (By specifying the proper category on the HEADER form you may subsequently delete a definition on its key form.) Display mode allows you to view the forms, not to change or add any information on them.

PHYSICAL DATABASE		NAME:	STAT:	
DESC:		(HDAM) - RMNAME:		MAXRBN:
DBACCS:		ANCH:		MAXLEN:
OSACCS:		(GSAM) - RECFM:		
PASSWD:		UPDATE:	DATMUSER-NO:	DATMUSER-SEQ:
SECURITY - QUERY:				

DSG	LABEL	DD1	BLK1	RECL1	DD2/OVFLW	BLK2	RECL2	DEV	MOD	SCAN	FSPF	FBFF
01												
02												
03												
04												
05												
06												
07												
08												
09												
10												


```

=====
ACT:  _  1-PROC  2-REGEN  3-DESC  4-ALIAS  5-STRUC          10-HDR  11-EXPLAIN
      6-USERDATA-NO:  _  9-REUSE
                               (RESPONSE LINE)
=====

```

DISPLAY Mode:

```

=====
ACT:  _          2-REGEN  3-DESC  4-ALIAS  5-STRUC          10-HDR  11-EXPLAIN
      6-USERDATA-NO:  _  9-REUSE
                               (RESPONSE LINE)
=====

```

DELETE Mode:

```

=====
ACT:  _  1-DELETE          9-REUSE
                               (RESPONSE LINE)
=====

```

Figure 4. Example of Display Form and Alternative Action Areas

THE ACTION AREA

The action area on the display form in Figure 4 is the one you see if you are working in edit mode. The actions available in delete or display mode may be different. The two separate action areas at the bottom of Figure 4 illustrate the delete and display mode actions. The alternative action areas are shown in this manner for each display form, when they exist, in Chapter 3.

ENTERING, MODIFYING, AND DELETING VALUES

Each form has a number of labeled fields or columns in which you can enter parameters for a given definition. When you first key in an entry on a display form, it is not highlighted, showing that it has not been stored in the Dictionary. Once you have submitted the information for processing by selecting the PROC action, the values are entered in the Dictionary, and redisplayed in highlight. You can enter any number of values on the form before submitting the entries for processing.

In general, you can modify the value of a stored parameter by writing over that value with the change you wish to make. The new value then replaces the old value in the Dictionary when the form is processed.

Similarly, you can delete an entry from the Dictionary by writing over the displayed entry with minus signs (-). This process is referred to as erasing the field. Again, any number of values can be modified or deleted before you submit the form for processing. Pressing the tab key moves the cursor from field to field.

DEFAULTS IN DISPLAY FORMS

The Dictionary is shipped with preset default values for subject name qualifiers on display forms. These values always appear on the HEADER form. If you wish to modify these values, you may do so on the HEADER form. To get the installation-defined default values, write over the values displayed with minus signs. To substitute your own values, write over the values displayed on the HEADER form.

ENTRIES IN TABLES

In most of the tables on the display forms, the values entered into the first column or group of columns serve as a key to the entries in all the columns for that line of the table. In these cases, you must enter a complete key before you can process the new entry. Further, you cannot delete any of the values comprising such a key without deleting the entire line or row.

In Figure 4, for example, the DSG (data set group) number in the first column is the key entry for all the entries in that numbered row. To delete the entire line, you just delete the DSG number.

The entries in specific tables are explained in detail in the descriptions of the display forms on which they appear.

PROC, REUSE, AND REGEN ACTIONS

You can make many changes to a display form in edit mode. You can enter values for any number of previously unspecified parameters and modify or delete current values for other parameters. To enter these changes into the Dictionary, you must select the PROC (process) action. If, after keying new entries or making changes or deletions, you select any other action, all the changes you made are lost.

Sometimes, you may want to enter, modify, or view a number of subjects belonging to a certain subject category without returning to the HEADER form. If you are using a key form, you may overlay the subject name (including subject name qualifiers) at the top of the current definition form with the new subject name and select the REUSE action. When the form is redisplayed the new subject and any stored data will appear in highlight. You may make changes and additions to this form and process it as you would any other.

In some cases, you may want to have the display form you are working on displayed as it was after the last change you processed. For example, you may discover errors in the changes you have keyed and want to start over without having to undo those changes. Selection of the REGEN (regenerate) action causes the display form to be displayed as it was before you made changes to it on the screen. Any changes that were processed before you selected REGEN will be unaffected.

At other times, you may suspect that another user has modified a Dictionary entry that you are working on. If you want to view the newly modified information on the display form, select the REGEN action.

PROCESSING A RESPONSE AND HIGHLIGHTING

After a form has been processed, it is redisplayed with a message (in the response line) telling you the results of the processing. If all your entries were processed successfully, you are so informed, and are free to continue entering and modifying values on the form. If, on the other hand, an error was encountered during processing, a diagnostic message is displayed.

When such an error occurs, some of your entries may have been successfully processed and others not. Multiple entries submitted for processing at once are in general processed from top to bottom and from left to right. Any entries that were processed and stored in the Dictionary before the error was detected are displayed in highlight. Entries that were not successfully processed (the error entry and subsequent entries) are displayed as they were before you selected PROC.

If you decide not to correct an error, but request another display form or select the REGEN action, the values not highlighted are lost.

SPECIAL CHARACTERISTICS OF DISPLAY FORMS

WINDOW FORMS

Certain display forms (those with a W below them in Figure 3) function as a window to be moved up and down over a list of entries too large for display in a single frame. For example, your Dictionary may include hundreds of entries for fields related to a given segment. The window on the SEGMENT-FIELDS form which displays these entries only has space for 15 related-field entries, but you may view all the stored entries by using the TOP and DOWN actions at the bottom of the form.

When the form is first displayed, the field with the lowest START value appears as the first line in the window. You can select the DOWN action to move the window down over the list; each selection of the DOWN action allows you to page through the list of entries. With each DOWN action a full window is displayed. If you prefer to step through the entries at a slower rate enter the number of lines the window is to be moved in the field adjacent to DOWN. Then select the DOWN action to move the window. To move to the end of the list, enter an asterisk (*) instead of a number in the field following DOWN. To return to the beginning of the list, select TOP.

Window forms can be used to delete, insert, and, in some cases, move entries in the stored lists.

EXPLANATION FORMS

Every operational display form is associated with one or more frames of explanatory text that explain that form and the labeled fields and columns it contains. To request an EXPLANATION form, select the EXPLAIN action, which appears on every form. The explanations appear in frames, with a sequence number at the top of each frame.

Three display forms display data for subjects in categories or relationships of a type defined by your installation (EXTENSIBILITY SUBJECT form, CATEGORY-RELATIONSHIPS form, and RELATIONSHIP DATA form). You may access installation-written explanations of the categories or relationships on these forms on an Extensibility EXPLANATION form. The form is described in Chapter 3. For information about how to access the Extensibility EXPLANATION form from those forms mentioned above, see the sections describing the forms in Chapter 3.

Any entries you made on the form on which you selected the EXPLAIN action will be lost when the EXPLANATION form is displayed.

SCREEN PROTECTION

The Dictionary protects certain information on the screen from being written over by inhibiting any input that would destroy a field or data presented for information only. To make moving around these protected fields easier, the tab key moves the cursor from field to field across and down the form.

THE STANDARD HEADER FORM

The standard HEADER form allows you to specify what task you need to perform. Under IMS/VS you can access the form by specifying the /FORMAT command with the HRDMOD or DISPMOD parameter or by selecting the HDR action on any other display form. You can access the form under CICS/VS by specifying the HRDMOD or DISPMOD parameter with the DICT MOD= transaction or by selecting the HDR action on any other form. (The operand DICT is an example and may be different at your installation, so check for the correct operand before using this one.) See the appropriate "Using the Dictionary Online" section earlier in this chapter for more information. The form is shown in Figure 5.

HRDMOD:

DB/DC DICTIONARY FACILITY					
CATEGORY	SUBJECT	CODE	OCC	STAT	
CATEGORY NBR/NAME: -SEGMENT IN DATABASE			000		
-RELATIONSHIPS TO:					
001-DATABASE	011-SYSDEF				
002-SEGMENT					
003-FIELD					
004-PCB					
005-SYSTEM					
006-PSB					
007-PROGRAM					
008-MODULE					
009-JOB					
010-TRANSACT					
DEFAULT QUALIFIERS - LANG: A OCC: 000 STAT: T DBTYPE: P					
=====					
ACT: _ 1-EDIT 2-DELETE 3-DISPLAY 4-QUIT 5-COMMAND			11-EXPLAIN		
(RESPONSE LINE)					

DISPMOD:

=====		
ACT: _	3-DISPLAY 4-QUIT	11-EXPLAIN
(RESPONSE LINE)		

Figure 5. HEADER Form

CONTENT

The standard HEADER form can be divided by function into three sections. The first section is used to specify what task is to be performed. The three rows at the top of the form are used to specify the categories and subject names necessary to perform the task. The three rows are divided into subfields by the five headings that appear along the top (CATEGORY, SUBJECT, CODE, OCC, and STAT). The rows and the subfields that they contain are explained below.

CATEGORY NBR/NAME

This row requires the category and subject name of the subject in which you are interested. You must always fill in this row, regardless of the kind of information you want to view or modify. The subfields for this row are described below.

- CATEGORY** The category number, name, or valid abbreviation of the category name for the subject specified in this row. For example, if the subject is a segment, category must be 2, SEGMENT, or any valid abbreviation for SEGMENT.
- SUBJECT** The user name portion of the subject name.
- CODE** The code qualifier for the name specified in the SUBJECT subfield.
- OCC** The occurrence number for the name specified in the SUBJECT subfield.
- STAT** The status code of the name specified in the SUBJECT subfield.

SEGMENT IN DATABASE

Fill in this row only if you have named a segment in the first row and wish to view or modify its relationship to a particular data base. Use this row to specify the subject name of the data base. The subfields for this row are described below.

- SUBJECT** The user name portion of the subject name of the data base.
- CODE** The CODE qualifier for the name specified in the SUBJECT subfield.
- OCC** The occurrence number for the name specified in the SUBJECT subfield. Since the subject in this row is always a data base, and a data base name must have an occurrence number of zero, this subfield is preset to 000.
- STAT** The status code for the name specified in the SUBJECT subfield.

RELATIONSHIPS TO

Put a category number, name, or valid abbreviation in this row only if you wish to view or modify a list of subjects related to the subject specified in the CATEGORY NBR/NAME row. The list of subjects produced will all be in the category specified in this row.

- CATEGORY** The category of the related subjects to be listed. Specify the category number, name, or valid abbreviation. You may use the values *SYO or *EXT in place of a category name in this row. For an explanation of these values, see "Working with the Form" below.

The second section of the standard HEADER form is a list of the numbers and names of the standard categories recognized by the Dictionary.

The last section of the form is a single row, just above the action area, that displays the default subject name qualifiers and their current values. On the first access of the HEADER form, the values displayed are the subject name qualifier default values for display forms shipped with the Dictionary. You may modify these values, and the values displayed on subsequent accesses of the HEADER form will reflect any modifications you have made. See Figure 8 for valid values.

WORKING WITH THE FORM

You use the HEADER form to specify what task you want to perform. At the same time you can choose whether the form on which you perform the task is to be displayed in edit, delete, or display mode. You also have the option of terminating your interactive session with the Dictionary via the QUIT action which is only available on the HEADER form.

You get to other forms by filling in fields at the top of the HEADER form and selecting one of the actions listed at the bottom. There are three kinds of tasks you may initiate on the forms you will access (excluding using the COMMAND form). You may define and modify subject definitions and associated data, relate a segment to a data base (or view that relationship), or view or modify the relationships between a subject and other subjects in a particular category. The processes for initiating these tasks vary. These processes and the forms they retrieve are described below.

Subjects in Categories

To retrieve information about any subject defined in the Dictionary, use the CATEGORY NBR/NAME row. Enter the category of the subject under the CATEGORY heading. You may use the category number or any valid abbreviation of the category name in place of the full name. Next enter the subject's user name and qualifier values under the appropriate headings in the same row. If you leave out any values, the default values for those qualifiers, which appear at the bottom of the form, will be used.

When you choose an action, the Dictionary will produce a key form. Which form depends on the category you have specified. Most categories lead to a specific form, specification of FIELD yields a FIELD form, PCB a PCB form etc. If you specify the category DATABASE, however, you may get any one of four forms, depending on the value you enter for the CODE qualifier.

To complete your request, you must choose an action. Select the EDIT, DELETE, or DISPLAY action according to what you want to do with the information on the form.

Segments in Data Bases

If you want information about the relationship between a segment and a data base, you must use the CATEGORY NBR/NAME row and the SEGMENT IN DATABASE row. Enter 2, SEGMENT, or any of its valid abbreviations under CATEGORY in the CATEGORY NBR/NAME row. Specify the user name and qualifier values for the segment under the appropriate headings in the same row. Next enter the user name and qualifier values for the data base the segment is to be related to under the appropriate headings in the SEGMENT IN DATABASE row. If the segment you specified has been defined previously, and related to the data base you specified with a particular role, one of the role-dependent segment forms will be displayed (LOG-SEGMENT, PHY-SEGMENT, PIP-SEGMENT, SIP-SEGMENT, or VLC-SEGMENT). Otherwise, any information available for this segment will be displayed on the no-role SEGMENT form. Use the EDIT or DISPLAY action to access the form in the mode appropriate to your use.

Related Subjects

To view or edit a list of subjects related to a particular subject, use the CATEGORY NBR/NAME row and the RELATIONSHIPS TO row. Specify the subject whose related subjects you wish to see, the principal subject, in the CATEGORY NBR/NAME row. Enter the category number, name, or abbreviation under the CATEGORY heading and the user name and qualifier values under the appropriate headings. To relate subjects to the principal subject, specify the category (related category) in which those subjects (related subjects) are to occur in the RELATIONSHIPS TO row. Enter a category number, name, or abbreviation under the CATEGORY heading in the RELATIONSHIPS TO row. To complete your request, use the EDIT or DISPLAY action.

The form you access depends on the two categories you enter in the two rows. If both are standard categories, the subsequent form is the RELATED ENTITIES form. In place of a standard category, you may specify *SYO in the RELATIONSHIPS TO row. *SYO causes related subjects in the seven system categories (SYSDEF, SYSTEM, JOB, PROGRAM, MODULE, TRANSACTION, and PSB) to be displayed on the RELATED ENTITIES form. If you wish to see related subjects in a specific system category, specify that category instead of *SYO.

If you specify an installation-defined category in either row or in both rows, the subsequent form is a selected SUBJECT-RELATIONSHIPS form. This form contains a list of subjects related to the principal subject that occur in the related category.

Regardless of the category of the principal subject, you may access a form that displays a list of any related subjects in all installation-defined categories. These related subjects are displayed on the unselected SUBJECT-RELATIONSHIPS form. To access the form, specify *EXT in the RELATIONSHIPS TO row under the CATEGORY heading.

Modifying Default Subject Qualifier Values

If you do not specify subject name qualifier values for the names in the CATEGORY NBR/NAME row and the SEGMENT IN DATABASE row, the Dictionary will automatically use the default values for subject name qualifiers currently displayed on the form. These values appear in highlight near the bottom of the form. If you want to modify these values, you may, by writing over them with alternate valid values. If you want the default values defined by your installation, you may get them by writing minus signs over the values provided by the Dictionary. The default values you specify will remain active until you either change them or terminate your interactive session with the Dictionary.

Processing Options

All the actions in the action area except QUIT cause a form to be displayed. Which form, depends on the values you have entered in the fields at the top of the form. The action you choose will determine in what mode the form will be presented (except for the QUIT and EXPLAIN actions). Select the action by writing the number of the action in the ACT field and pressing the ENTER key, or by pressing the PF key corresponding to the number of the action.

If you select the EDIT action (press PF key 1), the form that appears will be displayed in edit mode. You may alter information on the form, or if it is a new subject, enter the subject definition.

If you select the DELETE action (press PF key 2), the form that appears will be displayed in delete mode. You may use the DELETE action on that form to delete the subject (and its relationships to other subjects) from the Dictionary. You may not use the DELETE action when you have entered data in the SEGMENT IN

DATABASE or RELATIONSHIPS TO rows. If you do, an error message will be displayed.

If you select the DISPLAY action (press PF key 3), the form that appears will be displayed in display mode. You may not make any changes to the information on that form.

If you select the QUIT action (press PF key 4), you will terminate your interactive session.

If you select the COMMAND action (press PF key 5), the COMMAND form will be displayed. On the COMMAND form, you may enter Dictionary commands.

If you select the EXPLAIN action (press PF key 11), an explanation for the HEADER form is displayed. The explanation includes general information about the interactive display forms facility that is not included with any other explanations. Anything you entered on the form before you selected EXPLAIN remains unprocessed and is lost.

THE SECURITY HEADER FORM

The security HEADER form is the HEADER form you use when you have Dictionary security installed. It contains the same fields as the standard HEADER and additional fields for sign-on and security information. The SIGN-ON ID and PASSWORD fields at the top of the form are for your sign-on identification and password. The initial display also contains the fields STAT CODES (UPDT) and STAT CODES (VIEW). These fields are blank when the form is first displayed, but will contain the status codes of the subjects you may update and view when the form is redisplayed after your sign-on and password have been validated.

Unlike the standard HEADER form, the initial display of the security HEADER form does not contain the standard eleven Dictionary category numbers and names. However, you can enter a category number or name and a subject at the same time you enter your sign-on and password. Figure 6 shows the blank form.

When you limit your entries on the initial security HEADER form to your sign-on and password, the Dictionary checks your security authorization and redisplay the security HEADER form with whatever list of Dictionary categories your installation has decided to put there. Your access profile determines which categories you are able to access. If there is no blank line at the bottom of the frame, select the DOWN action to view subsequent frames of categories.

The form also displays the status codes you may update in the STAT CODES (UPDT) field. The status codes you may view only are displayed in the STAT CODES (VIEW) field. You may use the redisplayed security HEADER form the same way you would the standard HEADER form. Figure 7 shows a typical redisplayed security HEADER form.

The TOP and DOWN actions that appear on the redisplayed form allow you to page through the list of categories your installation has provided on the form.

Security Messages

In the course of using the Dictionary with security you may occasionally try to do something that is not permitted by your access profile. If you do, the Dictionary sends you an error message and does not complete your request. All messages are included in Appendix C at the end of this book.

HDRSEC:

DB/DC DICTIONARY FACILITY	SIGN-ON ID: PASSWORD:				
CATEGORY	SUBJECT		CODE	OCC	STAT
CATEGORY NBR/NAME: -SEGMENT IN DATABASE -RELATIONSHIPS TO:				000	
STAT CODES (UPDT): STAT CODES (VIEW): DEFAULT QUALIFIERS - LANG: A OCC: 000 STAT: T DBTYPE: P =====					
ACT: _ 1-EDIT 2-DELETE 3-DISPLAY 4-QUIT 5-COMMAND 11-EXPLAIN 6-DOWN: 8_ 7-TOP (RESPONSE LINE)					

DISPSEC:

=====					
ACT: _		3-DISPLAY	4-QUIT		11-EXPLAIN
	6-DOWN: 8_	7-TOP			
(RESPONSE LINE)					

Figure 6. Initial Security HEADER Form

HDRSEC:

DB/DC DICTIONARY FACILITY				
CATEGORY	SUBJECT	CODE	OCC	STAT
CATEGORY NBR/NAME: -SEGMENT IN DATABASE -RELATIONSHIPS TO:			000	
001-DATABASE	009-JOB			
002-SEGMENT	010-TRANSACTION			
003-FIELD	011-SYSDEF			
004-PCB				
005-SYSTEM				
006-PSB				
007-PROGRAM				
008-MODULE				
STAT CODES (UPDT): C F T				
STAT CODES (VIEW): C F P T				
DEFAULT QUALIFIERS - LANG: A OCC: 000 STAT: T DBTYPE: P				
=====				
ACT: _ 1-EDIT 2-DELETE 3-DISPLAY 4-QUIT 5-COMMAND 11-EXPLAIN				
6-DOWN: 8_ 7-TOP				
(RESPONSE LINE)				

DISPSEC:

=====				
ACT: _ 3-DISPLAY 4-QUIT 11-EXPLAIN				
6-DOWN: 8_ 7-TOP				
(RESPONSE LINE)				

Figure 7. Security HEADER Form After Sign-On

CHAPTER 3. DISPLAY FORMS

This chapter contains descriptions of the display forms. The standard and security HEADER forms are discussed in Chapter 2. The forms for defining and establishing relationships between subjects in installation-defined categories are included. Since the Extensibility Control Information (ECI) and DDUSER forms are designed for use by Dictionary administration personnel they are described in the DB/DC Data Dictionary Administration and Customization Guide.

DISPLAY FORM FIELDS

Many fields appear on more than one display form. Since these fields may have different uses on different forms, to attempt to summarize them all here would be confusing. A few of the fields occur on almost every form though. In order to decrease repetition of information, the common fields are discussed below. All other fields are discussed in the 'Contents' section of those forms on which they appear.

A subject name appears at the top of nearly every form. This subject name is broken down into four fields that represent the user name and its qualifiers.

NAME	User name portion of the subject name
CODE	Specifies type of category (language type for element or segment)
OCC	Occurrence number of the subject
STAT	Status of subject (production or test)

All of these fields may not occur at the top of every form, but the meaning of the fields is constant for all forms. Rules for the selection of qualifier values for display forms follow in Figure 8.

SUBJECT CATEGORY NAME (NBR) (ALTERNATIVE FORMS)	STATUS (See Notes 1&2) Default is T	SUBJECT NAME		
		SUBJECT CODE	USER NAME	OCCURRENCE NUMBER Default is 0
DATABASE (001) (D, DBS, FILE, L, FIL)	A-T or 0-9. All are test levels except P which is Production. Notes: 1. An alias has the same status as the name to which it belongs. 2. No more than two status codes in a structure. If there are two, one must be P.	P physical DL/I L logical, DL/I X primary index data base Y secondary index data base Default is P	1-8 A/N characters.	0 for DL/I
SEGMENT (002) (S, SEG, RECORD, R, REC)		A assembler language, DL/I B PL/I, DL/I C COBOL, DL/I J assembler language, non-DL/I K PL/I, non-DL/I L COBOL, non-DL/I Default is A	1-31 characters. Names with Subject code=A must be 1-8 A/N characters	0 to 255
ELEMENT (003) (E, DTE, FIELD, FLD)				
PCB (004) (P)		P PCB	1-8 A/N characters. Usually same as corresponding PSB.	1 to 255 (sequence of PCBs in PSB)
SYSDEF (011) (N, SDF)		N	1-31 characters.	0 to 255
SYSTEM (005) (Y, SYS)		S	1-31 characters.	0 to 255
JOB (009) (J)		J	1-8 A/N characters. Should follow operating system naming convention	0 to 255
PROGRAM (007) (G, PGM)		G		0 to 255
MODULE (008) (M, MOD)		M		0 to 255
TRANSACTION (010) (T, TRN, TRANSACT)		T		0 to 255
PSB (006) (B)	P	1-8 A/N characters. Should follow DL/I naming conventions	0	
Installation-defined (056-255)	none	1-31 characters.	0 to 255	

Figure 8. Subject Name Rules for Display Forms

Another field that occurs on most forms is the DESC field. This field can be used to enter one line (up to 72 characters) of freeform text describing the subject. Text entered into the DESC field is stored as line 001 of the description data associated with the subject at the top of the form. Any changes you make in the DESC field are reflected (after processing) in the stored description and in the displayed line 001 of the DESCRIPTION form for this subject. Similarly, changes made in line 001 on the DESCRIPTION form will appear in the DESC field here. If there is no text in line 001 of the description data, no text will appear in the DESC field.

DISPLAY FORM ACTIONS

The action area at the bottom of each form contains an ACT (action) field at the upper left corner, followed by one or two rows of actions. Each action is preceded by a number that uniquely identifies the processing option and the corresponding PF key for that display form. Some actions are followed by one or more fields for the entry of additional information about the action requested.

You can specify the action you want either by entering the appropriate number in the ACT field and then pressing the ENTER key or by pressing the function key with the same number as that action. If you have entered an action number in the action area and select a PF key instead of ENTER, the action indicated by the PF key takes precedence. If the action area contains a PROC action, pressing the ENTER key defaults to that action. In specifying actions that require additional information, enter the information before you press the ENTER key or the numbered function key.

The actions that are common to many forms are explained below. Actions that only occur on a specific display form are explained below the processing options section of that form's description.

ALIAS: This action displays the ALIASES form associated with the subject defined on the form from which this action is selected.

DELETE: This action appears on the HEADER form and on other forms. The HEADER form allows you to select delete mode. On the other forms, you can use the DELETE action to delete the Dictionary subject on the form.

DESC: This action displays the DESCRIPTION form associated with the subject defined on the form from which this action is selected.

DOWN: This action moves the window down over a stored list. The field adjacent to the DOWN action, the down count, specifies the number of lines the window will move for each selection of the DOWN action. You may modify the down count by writing over the default value. To move to the end of the list, place an asterisk in place of the down count and select the DOWN action. To return to the top of the list, select the TOP action.

EXPLAIN: This action displays the first of a set of frames of explanatory text associated with the display form from which this action is selected.

FLDS: This action displays the SEGMENT-FIELDS form. The form displays the fields related to the segment that is the subject of the form from which this action is selected.

HDR: This action returns to the HEADER form, from which you can select another key form or select the QUIT action.

PROC: This action causes any entries, modifications, or deletions made by the user in labeled fields or columns to be processed by the Dictionary. New entries are highlighted after successful processing.

REGEN: This action regenerates the contents of the form from the stored Dictionary contents, eliminating any changes entered but not yet processed, and including any changes made in the stored contents from some other source (another display station) while this form has been displayed.

REUSE: This action allows the same form to be used to display another subject in the same category without returning to the HEADER form. Generally, changing the subject name at the top of the form and selecting the REUSE action displays the same kind of form showing the appropriate new data for the new name specified. Use of the REUSE action differs on some forms so, consult the "Working with the Form" section for the individual form before selecting the REUSE action.

RTN: This action returns to the form one level up in the forms hierarchy; for example, a return to the PHYSICAL DATABASE form from the subordinate PHYSICAL DATABASE STRUCTURE form.

TOP: This action moves the window to the beginning (the first entry or first line) of the stored list.

USERDATA-NO: This action permits you to select a set of user data for the subject of the definition form.

The display forms are described in alphabetic order below:

ALIASES FORM

The ALIASES form is a table in which up to 14 aliases can be entered and displayed for a given subject in the data base, segment, PCB, element, or any installation-defined categories. The Dictionary does not place any limit on the number of aliases stored for these subjects, but only 14 can be displayed or entered on the ALIASES form. If more than 14 aliases exist, a message appears in the response line the first time the form is displayed.

Since an alias is assumed to have characteristics identical to those of the primary name, an alias entry consists of just the alias subject name.

The blank display form in edit mode is shown in Figure 9.

Accessing the ALIASES Form

You can access the ALIASES form by selecting the ALIAS action for a given subject from the following display forms:

EXTENSIBILITY SUBJECT	PHY-SEGMENT
FIELD	PIP-SEGMENT
LOGICAL DATABASE	no-role SEGMENT
LOG-SEGMENT	SECONDARY INDEX DATABASE
PCB	SIP-SEGMENT
PHYSICAL DATABASE	VLC-SEGMENT

Content

The name of the subject on the form from which you requested a list of aliases is displayed in appropriately labeled fields at the top of the ALIASES form. This name may be a primary name or an alias. This name cannot be changed on this form. The table that makes up most of the form has four columns, headed NAME, CODE (subject code), OCC (occurrence number), and STAT (status). You can enter the aliases after the primary name.

The primary name of the subject appears as the first name in the table.

ALIASES	category:	CODE:	OCC:	STAT:
NAME		CODE	OCC	STAT
*** PRIMARY NAME ***				
===== ACT: _ 1-PROC 2-REGEN 9-RTN 10-HDR 11-EXPLAIN (RESPONSE LINE)				

DISPLAY Mode:

ACT: _	2-REGEN	9-RTN	10-HDR	11-EXPLAIN
(RESPONSE LINE)				

Figure 9. ALIASES Form

Working with the Form

If one or more empty lines appear in the table, you can enter alias names in these lines. Note that each alias name must be unique within a category. If you attempt to enter an alias that already exists in a category, or if you attempt to enter an alias that is already a stored name with associated attributes, an error message will be displayed in the response line.

An alias does not have to have the code and occurrence number as the primary subject, but it must have the same status. If no entries are made in the CODE, OCC, or STAT columns, the alias entered is given the current default values.

Once an alias has been entered into the table, it cannot be modified. To delete an entry from the list and the Dictionary, erase the NAME portion of the alias by writing over it with minus signs, and then select PROC.

Processing Options

The options available for this form are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
9-RTN	returns to higher level form
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

CATEGORY-RELATIONSHIPS FORM

The CATEGORY-RELATIONSHIPS form is a window form that displays a list of relationship keywords that may be used with the installation-defined category named on the form. The keywords describe the types of relationships that subjects in that category may participate in. The form is not used to enter data or establish relationships. The blank form is shown in Figure 10.

CATEGORY-RELATIONSHIPS		CATEGORY:				
RELATIONSHIP	TO-CATEGORY	INV-KEYWORD	SEQOPT	SEQATTR	DIRECTED	MARK
=====						
ACT: _		3-TOP	6-DOWN: 16_	8-EXPREL	9-RTN	10-HDR 11-EXPLAIN
(RESPONSE LINE)						

Figure 10. CATEGORY-RELATIONSHIPS Form

Accessing the CATEGORY-RELATIONSHIPS Form

You can access this form from a selected or unselected SUBJECT-RELATIONSHIPS form by selecting the CAT-RELATIONSHIPS action.

Content

The top of the form displays the principal category of interest; the subject residing in this category was originally shown at the top of the SUBJECT-RELATIONSHIPS form.

The body of the form is a table. The first two columns, headed RELATIONSHIP and TO-CATEGORY, list each relationship keyword and the name of the category to which subjects in the principal category may be related. In the next column, the inverse keyword (INV-KEYWORD) for the relationship is displayed to give a sense of direction to the relationship.

The column headed SEQOPT indicates when a relationship is sequenced. If the relationship is sequenced, the letter Y appears; otherwise, the column entry is blank.

The column headed SEQATTR contains the name of the relationship-data attribute that controls the sequence. Entries in this column only occur when the relationship is sequenced.

The entries in the DIRECTED column help you interpret the structural nature of the relationship. The principal category and the related categories shown in the TO-CATEGORY column may have a relationship with a distinct sense of dependency. This is indicated by a value in the DIRECTED column. The meaning of the values that can occur in the DIRECTED column are:

- D The relationship is downward; subjects residing in the TO-CATEGORY are subordinate to subjects in the principal category.
- U The relationship is upward; subjects residing in the TO-CATEGORY are superior to those in the principal category.
- N The relationship is not directed.

The MARK column is used in conjunction with the EXPREL action, and allows you to "mark" a relationship entry.

Working with the Form

This form only displays information; no changes or additions can be made in the displayed relationships. All areas except the MARK column are protected.

The MARK column allows the entry of an * as a mark on a single relationship. When you select the EXPREL action, it causes the explanation provided by your installation for the particular relationship to be displayed on the EXPLANATION form. The EXPREL action works with a single row. The topmost * in the MARK column is used.

The displayed entries in the SEQOPT and SEQATTR columns taken with the information gathered by viewing the EXPLANATION form should be recorded so that the appropriate entries can be made on the SUBJECT-RELATIONSHIPS form. The keyword in the RELATIONSHIP column is the one entered on the SUBJECT-RELATIONSHIPS form. Of special interest is whether a relationship is directed, and for the sequencing attribute (SEQATTR), the data type or value restrictions.

The RTN action causes a return to either of the two SUBJECT-RELATIONSHIPS forms, depending on which was displayed last.

Processing Options

The options available for this form are:

- 3-TOP positions window at top of stored list
- 6-DOWN moves window down over stored list
- 8-EXPREL displays the Extensibility EXPLANATION form
- 9-RTN returns to previously displayed SUBJECT-RELATIONSHIPS form
- 10-HDR returns to HEADER form
- 11-EXPLAIN displays the EXPLANATION form

Enter the action number in the ACT field or select a PF key.

Specialized Actions

EXPREL: This action displays the Extensibility EXPLANATION form which is provided by your installation to describe an installation-defined relationship and its relationship data.

COMMAND FORM

The COMMAND form is a specialized display form you use if you want to enter commands during a terminal session. It allows you to enter commands on the form and displays the output created from the command you entered.

If you want to use the COMMAND form, select the COMMAND action from the HEADER form. The COMMAND form can be accessed at any time during the terminal session. The blank form is shown in Figure 11.

DB/DC DICTIONARY COMMAND FORM

=====

ACT: _ 1-PROC PA1=NEXT PAGE 10-HDR 11-EXPLAIN

Figure 11. DB/DC Dictionary COMMAND Form

Content

The first ten lines under the form title are available for a single command. The response to the command will use lines remaining below the command.

Working with the Form

Any attempt to enter data beyond the tenth line will be rejected. Any line following a command input line which does not have a continuation sign (a + sign) will be ignored.

The command entered is retained so that corrections or modifications may be made.

The response to the command will begin on the line directly under the last line of the command. If the response cannot be fully contained on the form, depressing the PA1 key causes the screen to be cleared and filled with more of the response. Continue to press the PA1 key until you have viewed all the output. See "Output from Online Operations" in Chapter 1 for both IMS/VS and CICS/VS for more information.

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form
PA1	displays next page (See note below)

Enter the action number in the ACT field, or select a PF key.

Note: If you are using this form with CICS/VS, enter A1 instead of using the PA1 key for the same effect.

DESCRIPTION FORM

The DESCRIPTION form is a window form that is used to enter or display description data. Line number 001 of description text is shown at the top of most of the key forms for Dictionary subjects. The format in edit mode is shown in Figure 12.

Accessing the DESCRIPTION Form

You can access the DESCRIPTION form by selecting the DESC action on any of the following forms.

CAT-RELATIONSHIPS	PIP-SEGMENT
EXTENSIBILITY SUBJECT	PRIMARY INDEX DATABASE
FIELD	PROGRAM
JOB	PSB
LOGICAL DATABASE	SECONDARY INDEX DATABASE
LOG-SEGMENT	SIP-SEGMENT
MODULE	SYSDEF
no-role SEGMENT	SYSTEM
PCB	TRANSACTION
PHY-SEGMENT	VLC-SEGMENT
PHYSICAL DATABASE	

Content

The subject is displayed in appropriately labeled fields at the top of the form and cannot be modified on this form. The table that makes up most of the form is a window on the lines of text making up the description. The table has just two columns: LNO (line number) and TEXT. On the text part of the line you may enter freeform descriptive text. There is no restriction on the format or content of the text, but each line must have a unique number (from 001-999), and the line may not exceed 72 characters. This form does **not** allow you to modify the line number.

Working with the Form

When the DESCRIPTION form is first presented to the user, the window is positioned at the top of the list of lines; that is, on the first 15 line positions. A line of text with line number 001 is the same text that is in the description field on each key form, and any change in that line for a given subject is reflected on both the DESCRIPTION form and in line 001 on the key form.

Successive lines of text need not be assigned sequential line numbers; in fact, it is a good idea to skip line numbers to allow for future inserts. But the description text is always displayed in line-number order on the DESCRIPTION form and on printed reports.

To enter an additional line or lines of descriptive text, first position the window to obtain blank lines at the bottom of the display frame. (If no description has ever been entered for the

DESCRIPTION	category:	CODE:	OCC:	STAT:
LNO	TEXT			
===== ACT: _ 1-PROC 2-REGEN 3-TOP 9-RTN 10-HDR 11-EXPLAIN 6-DOWN: 15_ (RESPONSE LINE)				

DISPLAY Mode:

===== ACT: _ 2-REGEN 3-TOP 9-RTN 10-HDR 11-EXPLAIN 6-DOWN: 15_ (RESPONSE LINE)				
--	--	--	--	--

Figure 12. DESCRIPTION Form

subject in question, the table portion of the display form will be entirely blank.) You may get to blank lines by specifying * in place of the down count and selecting the DOWN action. Then enter the additional line or lines of text with their line numbers in the appropriate columns.

After checking your entry for accuracy, select the PROC action. The new lines will then be moved to the positions within the stored text indicated by their line numbers. (They may disappear from the window currently displayed.)

To delete a line from the Dictionary, first position the window to display the line. Then write over the line number with minus signs, and select PROC. To replace a current line, modify the line as displayed, and select PROC.

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
3-TOP	positions window at top of stored list
6-DOWN	moves window down over stored list
9-RTN	returns to higher level form
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

EXPLANATION FORM

Every display form has associated with it an EXPLANATION form explaining its use. You access the EXPLANATION form by selecting the EXPLAIN action in the action area of any form. The EXPLANATION form is shown in Figure 13. This form may include one or more frames of explanation.

EXPLANATION	(subject frame name)
=====	
ACT: _	2-NEXT 3-PREVIOUS 9-RTN 10-HDR (RESPONSE LINE)

Figure 13. EXPLANATION Form

Content

The name of the display form that is being explained and the sequence number of the EXPLANATION form appear at the top of the EXPLANATION form. The rest of the form contains text.

Working with the Form

The explanation displayed should not be modified; modifications are not processed.

Processing Options

The options available for this form are:

2-NEXT	displays next frame of explanation
3-PREVIOUS	displays previous frame of explanation
9-RTN	returns to display form being explained
10-HDR	returns to HEADER form

Enter the action number in the ACT field, or select a PF key.

EXTENSIBILITY EXPLANATION FORM

Each installation-defined category may have defined for it explanation text that describes the subjects in that category and the requirements for entering data. The forms which display installation-defined relationship-types also have explanation text about those relationship-types associated with them. You can request the explanation by selecting EXPLAIN from the action choices offered. A blank Extensibility EXPLANATION form is shown in Figure 14.

EXPLANATION	(explanation subject identification)
=====	
ACT: _	3-TOP 6-DOWN: 18_ 9-RTN 10-HDR 11-EXPLAIN
(RESPONSE LINE)	

Figure 14. Extensibility EXPLANATION Form

Content

The name of the installation-defined category or relationship-type appears at the top of the EXPLANATION form. The rest of the form contains text.

Working with the Form

The explanation displayed may not be modified; modifications are not processed.

Note that the action area is different from that of the EXPLANATION form. You move the window down over the text by selecting the DOWN action. To view a prior segment of text, you return to the top and page back through the explanation.

Processing Options

The options available for this form are:

3-TOP	displays the first frame of explanation
6-DOWN	moves window down over stored text
9-RTN	returns to display form being explained
10-HDR	returns to HEADER form
11-EXPLAIN	displays the EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

EXTENSIBILITY SUBJECT FORM

This form is used to enter and display subject data for a subject in an installation-defined category. The form has a unique title corresponding to the name of the installation-defined category. (This name replaces the x's at the top of the form.) Figure 15 shows a blank form in edit mode.

category	NAME:	
	OCC:	STAT:
DESC:		
=====		
ACT: _	1-PROC	2-REGEN 3-DESC 4-ALIAS 9-REUSE 10-HDR 11-EXPLAIN
	7-UP: _	8-DOWN: _ 6-USERDATA-NO: _ 5-RELATIONSHIPS: _____
(RESPONSE LINE)		

DISPLAY Mode:

=====		
ACT: _	2-REGEN 3-DESC 4-ALIAS	9-REUSE 10-HDR 11-EXPLAIN
	7-UP: _	8-DOWN: _ 6-USERDATA-NO: _ 5-RELATIONSHIPS: _____
(RESPONSE LINE)		

DELETE Mode:

=====		
ACT: _	1-DELETE	10-HDR 11-EXPLAIN
(RESPONSE LINE)		

Figure 15. EXTENSIBILITY SUBJECT Form

Accessing the EXTENSIBILITY SUBJECT Form

You can access this form from the HEADER form, from either of the SUBJECT-RELATIONSHIPS forms, or from a RELATIONSHIP DATA form.

Content

The top of the form contains the name and qualifiers of the subject. The body of the form is a table of keywords and any stored values that exist. Keywords with accompanying values are listed downward in the left side of the table, and continue on the top right side. A keyword's value may be up to 120 characters long. There is a fixed value field associated with each keyword. That field may be from one to five 24 character lines in the value column. A keyword and its value always appear in the same column; so if there is not enough room at the bottom of a column for the value field, the keyword and its value will appear in the next column or on the next screen.

Working with the Form

To view additional keywords and values on the form, use the UP and DOWN actions. You may specify a count in the field adjacent to both actions. The count you specify refers to the number of keywords the window will move over.

The presence of one or more blank lines at the bottom of the right-hand column of keywords does not necessarily mean you have reached the end of the list. It would be a good practice to look beyond the apparent end of the list by using the DOWN action.

If a keyword's value cannot be specified completely in the 24 positions available to each keyword line, it is continued in the next row in the Value column. Up to four continuation lines may be added. When a value does require a continuation, the value must be entered totally within the rows allotted for that keyword. The value can NOT have embedded blanks, so that its characters extend to the column boundary, and are continued in the first available position in the succeeding line.

To delete values, erase the whole value by overwriting it with minus signs. To change values, write over the old data and blank out any extraneous characters. Values that you add or change will reappear after processing just as you entered them. If you select the REGEN action, the values will be reformatted and left justified in the column.

Values you enter or modify are validated by the Dictionary and checked for a variety of characteristics that depend on the keyword. The value may be checked for length, to see if it occurs in a specified range, to see if it is a member of a list, or in some other way by an installation-written routine. If a keyword value is invalid, an error message will appear in the response line.

If your installation has written explanation text for the category, you may access it by selecting the EXPLAIN action, which displays an Extensibility EXPLANATION form. The text will typically describe the keywords and list valid values.

You can select the ALIAS action to define aliases for the subject.

You can use the RELATIONSHIPS action to display a list of subjects related to the subject on this form. To access this list, specify a category name in the field adjacent to the RELATIONSHIPS action. You may specify a standard or an installation-defined category. In either case, the list will be displayed on a selected SUBJECT-RELATIONSHIP form. If you do not specify any category in the field, the unselected SUBJECT-RELATIONSHIP will be displayed. It contains a list of any relationships established between the subject on the EXTENSIBILITY SUBJECT form and subjects in all installation-defined categories.

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of form
3-DESC	displays DESCRIPTION form for subject
4-ALIAS	displays ALIASES form for subject
5-RELATIONSHIPS	displays SUBJECT-RELATIONSHIPS form
6-USERDATA_NO	displays USER DATA form
7-UP	moves window up over list of of attributes for subject
8-DOWN	moves window down over list of attributes for subject
9-REUSE	redispays form for use with another subject
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

Specialized Actions

RELATIONSHIPS: This action displays the selected or unselected SUBJECT-RELATIONSHIPS form. If a category name is specified, the selected SUBJECT-RELATIONSHIPS form is displayed. This form contains a list of subjects, in the category specified on the previous form, that are related to the subject specified on the previous form.

If no category is specified, the unselected SUBJECT-RELATIONSHIPS form is displayed. This form contains a list of subjects related to the subject on the EXTENSIBILITY SUBJECT form in any installation-defined category.

UP: This action moves the window up over the list of attributes for the subject.

FIELD FORM

The FIELD form is used to enter and display the definition for a field or a subject in the element category. In addition to the standard DL/I field specifications, the form includes special COBOL, PL/I, and assembler language parameters. The blank form in edit mode is shown in Figure 16.

Accessing the FIELD Form

You can access the FIELD form from the HEADER form.

Content

The NAME, CODE, OCC, and STAT fields at the top of the form contain the user name and qualifiers for the field. You cannot modify the values specified in these fields. Subject names with a code of A (DL/I names) are truncated to a maximum of 8 alphameric characters for certain types of output processing.

The meaning of the fields is as follows:

DATE

This field specifies any date you want to associate with the defined field. Valid values must be six digits long in the form, "mmddy". Values may range from 010100 to 123199.

BYTES

This field specifies the field length in bytes. The value may be 1 to 32767.

FIELD	NAME:								
	CODE:	OCC:	STAT:						
DESC:									
DATE:	BYTES:	TYPE:	DIGITS:	DECIMALS:					
(COBOL & ASSEMBLER DATA)-									
USAGE:	SIGN:	BLANK:	JUST:	SYNCH:					
PICTURE:									
OCCURS:									
VALUE:									
(PL/I DATA)-									
PLIFMT:	SCALE/OPT:	MODE:	ALIGNED:						
DIM1: (:) DIM2: (:) DIM3: (:)			
PLIPIC:									
INITIAL:									
=====									
ACT: _	1-PROC	2-REGEN	3-DESC	4-ALIAS	5-SEGS	7-PL/I	10-HDR	11-EXPLAIN	
	6-USERDATA-NO: _	8-RELATED FIELDS: _	9-REUSE	13-INQATTR					
									(RESPONSE LINE)

DISPLAY Mode:

=====									
ACT: _	2-REGEN	3-DESC	4-ALIAS	5-SEGS	7-PL/I	10-HDR	11-EXPLAIN		
	6-USERDATA-NO: _	8-RELATED FIELDS: _	9-REUSE	13-INQATTR					
									(RESPONSE LINE)

DELETE Mode:

=====									
ACT: _	1-DELETE					10-HDR	11-EXPLAIN		
						9-REUSE			
									(RESPONSE LINE)

Figure 16. FIELD Form

If the value of PLIFMT is B (bit string), then the value of this field specifies a length in bits.

TYPE

This field specifies the type of data in the field. Valid values are:

- B** binary digits
- C** characters
- D** long floating point
- E** short floating point
- F** fixed point binary (word)
- H** fixed point binary (halfword)
- P** packed decimal
- X** hexadecimal digits
- Z** zoned decimal

DIGITS

This field specifies the minimum number of significant digits to be maintained for the data element. The value also represents the number of positions to be reserved for this field's data in reports. In PL/I, this value is the PRECISION attribute of an element. The value may be 1 to 255.

DECIMALS

This field specifies the scaling factor to be applied to the position of the decimal point for the data element. In PL/I, this value is interpreted as the SCALE attribute for an element. Possible values are -127 to +127.

(COBOL & ASSEMBLER DATA)-

The parameters specified in the COBOL and ASSEMBLER DATA fields pertain to COBOL and assembler language declarations for the field. Note that OCCURS and VALUE fields can each be up to 120 characters long (two lines of 60 characters each); a PICTURE field can have a maximum of 30 alphanumeric characters. The VALUE field must be enclosed by single quotation marks when it represents an alphanumeric value. If it represents a numeric value, it must be entered without quotation marks. (A displayed numeric value (USAGE=D), however, must be enclosed in single quotation marks.)

USAGE

This field specifies the data for the COBOL USAGE clause. The one-character value can be:

0	comp
1	comp-1
2	comp-2
3	comp-3
4	comp-4
D	display
I	index

SIGN

This field specifies the use of the COBOL SIGN clause. Valid characters are:

L	sign leading
LS	sign leading and separate clause
T	sign trailing
TS	sign trailing and separate clause

BLANK

This field specifies whether the field has a COBOL BLANK WHEN ZERO clause. A value of B specifies that the field has a BLANK WHEN ZERO clause. If the value is omitted, there is no clause.

JUST

This field specifies whether the field has a COBOL JUSTIFIED clause. A value of R specifies that the field has a COBOL JUSTIFIED clause. If the value is omitted, there is no clause.

SYNCH

This field specifies the use of the COBOL SYNCHRONIZE clause. Valid characters are:

S synchronize
SL synchronize left
SR synchronize right

PICTURE

This field permits the specification of a COBOL PICTURE attribute for an element. You may enter up to 30 characters, but these must include opening and closing single quotation marks.

Special consideration must be given to the values entered for OCCURS and VALUE: specifications requiring more than one line on the screen (60 characters) must be split between the two lines as if they were contiguous. This means that words may be split between the last position of the first line and the first position of the second as if the two lines were one. Also, in the VALUE specification, the opening quotation mark may appear on the first line and the closing quotation mark on the second line. However, when attempting to change a specification, the two lines must be thought of as separate fields. This means that to change a two-line specification to one that will fit on one line, the data on the second line must be erased (written over with minus signs) before you select the PROC action.

OCCURS

This field permits the specification of a COBOL OCCURS attribute for an element. You may enter up to 120 characters, but these must include opening and closing single quotation marks. See the paragraph above for more information.

VALUE

This field permits the specification of a COBOL VALUE attribute for an element. You may enter up to 120 characters, but these must include opening and closing single quotation marks. See the paragraph above for more information.

(PL/I DATA)-

The bottom half of the FIELD form is reserved for attributes of a PL/I data element or field:

PLIFMT

This field specifies the internal format of a PL/I variable. Valid characters are:

B bit string
C character string
D decimal value
E binary value
P pointer variable

SCALE/OPT

This field is used to indicate the scale of an arithmetic variable or a varying-length string variable. Valid characters are:

L floating point attribute
X fixed point attribute
V varying attribute

MODE

This field indicates the mode of either a real or a complex number. Specify R or C, respectively.

ALIGNED

This field enables the ALIGNED attribute to be specified for a data element. Enter the Y (yes, aligned) or N (no, unaligned).

DIM1 <DIM2> <DIM3>

These fields allow one to three dimension attributes to be specified for the element. Each dimension specification can be given as a pair of lower- and upper-bound values.

The full range of values for the bounds of the dimension is -32767 to +32767. If only an upper bound is given, it must be a positive, non-zero number. In this case the lower bound defaults to 1.

PLIPIC

This field permits the specification of a PL/I PICTURE attribute for an element. You may enter up to 30 characters, but these must include opening and closing single quotation marks.

INITIAL

This field permits an initial value to be specified for the element. You may enter up to 30 characters, but these must include opening and closing parentheses. String values must also be enclosed in single quotation marks, and two single quotation marks (') must be used to represent a single quotation mark within a character string.

Working with the Form

If you modify the values specified in the NAME, OCC, and STAT fields at the top of the form, you are expected to select the REUSE action. In this event, changes to other parts of the screen will be lost.

To define inquiry attributes with the INQUIRY ATTRIBUTES form, enter 13 into the field adjacent to ACT and process.

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
3-DESC	displays DESCRIPTION form for subject
4-ALIAS	displays ALIASES form for subject
5-SEGS	displays FIELD-SEGMENTS form
6-USERDATA_NO	selects set of User Data for subject
7-PL/I	displays PL/I DATA form for field
8-RELATED FIELDS	displays SUBORDINATE/SUPERIOR FIELDS form
9-REUSE	redispays form for use with another subject
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form
13-INQATTR	displays INQUIRY ATTRIBUTES form

Enter the action number in the ACT field or select a PF key.

Specialized Actions

INQATTR: This action displays the INQUIRY ATTRIBUTES form. This form contains the characteristics that a field or data element may possess in the context of inquiry systems or data base reports. These characteristics include report header text and parameters that describe what edit or validation actions are to occur.

RELATED FIELDS: This action displays the SUBORDINATE/SUPERIOR FIELDS form. This form displays a list of fields related to the field that was specified on the previously selected FIELD form. When you choose this action, you have to indicate in the field following the action whether you want to see fields superior (SUP) or subordinate (SUB) to the subject field.

SEGS: This action displays the FIELD-SEGMENTS form. This form displays a list of segments related to the field that is the subject of the previously selected FIELD form.

FIELD-SEGMENTS FORM

The FIELD-SEGMENTS form is a window form that documents the segments to which a subject field is related. On this form you can only enter relationships between previously defined fields and segments. The blank form in edit mode is shown in Figure 17.

Accessing the FIELD-SEGMENTS Form

You can access the form by selecting the SEGS action on the FIELD form.

FIELD-SEGMENTS		FIELD:						
		CODE:	OCC:	STAT:				
SEGMENT		CODE	OCC	STAT	START	BST	SEQ/GEN	
<p>=====</p> <p>ACT: _ 1-PROC 2-REGEN 3-TOP 9-RTN 10-HDR 11-EXPLAIN</p> <p>6-DOWN: 15_</p> <p>(RESPONSE LINE)</p>								

DISPLAY Mode:

<p>=====</p> <p>ACT: _ 2-REGEN 3-TOP 9-RTN 10-HDR 11-EXPLAIN</p> <p>6-DOWN: 15_</p> <p>(RESPONSE LINE)</p>								
--	--	--	--	--	--	--	--	--

Figure 17. FIELD-SEGMENTS Form

Content

The FIELD, CODE, OCC, and STAT fields at the top of the form contain the full subject name for this field. The subject name cannot be modified on this form. The table has seven labeled columns. The first four (SEGMENT, CODE, OCC, and STAT) are for the subject names of the segments listed. (See Figure 8 for subject rules.) The remaining three are for entries as follows:

START

This field specifies the starting position of the field in the segment. The value can be 1 to 5 numeric characters with a range of 1 to 32767. The default is 1.

BST

This field specifies the starting bit within the starting byte where this field begins. The value can be 1 to 8. This field is only valid for PL/I and assembler. It is not valid for COBOL.

SEQ/GEN

This field specifies whether the field contains a unique value, or it specifies whether the field appears with the segment in DBD_OUT. Valid values are:

null	no
U	unique values
M	multiple values
G	yes

One column is provided for a SEQ or GEN entry, since sequence fields appear with the segment in DBD_OUT.

Working with the Form

When the form is first displayed, the window is positioned at the top of the list of segments related to the field. Use the DOWN action to page through the list. To return to the beginning of the list, select TOP.

To enter the names of additional segments to which the field is related, first position the window so that one or more blank lines appear at the bottom of the form. Only the subject has to be entered (and be processed) to create an entry. If not all the subject qualifiers are entered, the current defaults are assumed.

On this form, you can only enter relationships for segments already defined in the Dictionary. If the subject you enter is not already in the segment category, you will receive an error message.

Once entered into the table, a segment subject cannot be modified on this form. An entry can be deleted from the table by erasing (writing over with minus signs) the user name (SEGMENT column) portion of the subject name. On processing, the relationship between the specified segment and the subject field is deleted from the Dictionary, but the field and segment definitions are not affected. Any or all of the segment entries displayed on the form at a given time can be deleted in this way.

The values specified in the START, BST, and SEQ/GEN fields are considered to be parameters of the field-segment relationship (that is, relationship data) and can be modified at your discretion.

Processing Options

The options available for this form while in edit mode are:

- 1-PROC processes entered data
- 2-REGEN regenerates contents of current form
- 3-TOP positions window at top of stored list
- 6-DOWN moves window down over stored list
- 9-RTN returns to previously displayed FIELD form
- 10-HDR returns to HEADER form
- 11-EXPLAIN displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

INQUIRY ATTRIBUTES FORM

The INQUIRY ATTRIBUTES form is used to specify the characteristics that a field or data element may possess in the context of inquiry systems or data base reports. These characteristics include report header text, and parameters that describe what edit or validation actions are to occur. The blank form in edit mode is shown in Figure 18.

INQUIRY ATTRIBUTES		NAME:		OCC:		STAT:			
(SECURITY) - QUERY:		UPDATE:							
HEADER-LINES:		HEADER-LENGTH:							
HEADER-TEXT:									
MASK:									
L K U P	E X I I G T	R N I C E	P I T	ROUTINE	TABLE NAME	ARGUMENT OR FUNCTION UNITS LENGTH	ERROR ACTION	EDITUSER NO SEQ	
DECODE		ENCODE		VALIDATE					
=====									
ACT: _		1-PROC 2-REGEN		(RESPONSE LINE)		9-RTN		10-HDR 11-EXPLAIN	

DISPLAY Mode:

=====									
ACT: _		2-REGEN		(RESPONSE LINE)		9-RTN		10-HDR 11-EXPLAIN	

Figure 18. INQUIRY ATTRIBUTES Form

Accessing the INQUIRY ATTRIBUTES form

You can access this form by specifying the INQATTR action on the FIELD form.

Content

The NAME, STAT, OCC, and CODE fields at the top are carried over from the FIELD form and cannot be altered. The upper portion of the form is used for header information. The lower portion has to do with edit outlines.

(SECURITY)

QUERY

This field specifies an access code for read-only access of a field in a data base. Any three alphameric characters are valid.

If GIS data base security is enabled, values may be numeric only, from 0 to 128. The value 0 means there is no security access code required.

UPDATE

This field specifies an access code for authorization to update a field in a data base. Any three alphameric characters are valid.

If GIS data base security is enabled, values may be numeric only, from 0 to 128. The value 0 means no security access code is required.

HEADER-LINES

This field specifies the number of lines into which the header text is to be separated.

HEADER-LENGTH

This field specifies the length (number of characters) of each of the header lines.

HEADER-TEXT

This field specifies the text to be presented with the subject field in a report. One or two lines of 60 characters each can be entered on the form.

MASK

This field specifies an edit mask for a numeric field. The values entered are strings of edit mask control characters.

DECODE, ENCODE, VALIDATE

Enter on the appropriate line the parameters to be applied to the method of processing the subject inquiry field. DECODE is for processing prior to presentation in a report. ENCODE and VALIDATE are for processing to be carried out prior to storage of the field value in the data base.

The first four positions specify the type of edit to be performed. The edit types are displayed above four columns. To select an edit, place an x in the appropriate column. The types of edits available are described below.

LKUP	Perform table lookup
EXIT	Process by means of a user exit
RNGE	Perform range validation
PICT	Perform a PICTURE edit

ROUTINE

If EXIT was specified at the beginning of the line, this field may be used to enter the name of the user exit routine. Enter up to eight characters.

TABLE NAME

If LKUP was specified at the beginning of the line, this field may be used to enter the table name. Enter up to eight characters.

UNITS, LENGTH

These fields may describe the type and length of data to be processed as input to an encode or validation operation, or they may specify attributes of the output from a decode operation.

Valid characters for UNITS are:

C Data is a string of characters
P Data is packed decimal
Z Data is zoned decimal

Valid values for LENGTH are 1 to 255.

ERROR ACTION

This field describes the action to be taken if a validation or encoding operation fails.

For GIS, two actions are defined:

E Prevent the field from being updated
S Replace the data with a default value, zeros or blanks, as appropriate

EDITUSER

Enter on the line under the heading a pair of values (NO, SEQ) that specify the User Data segment number and starting line number. Valid values for NO are 1,2,3,4, or 5. Line numbers must be from 1 to 999. The User Data lines must contain a pair of range values, a picture specification, or a block of lines forming a lookup table.

Working with the Form

The fields that make up a header line have special application to GIS queries. Each header text line is padded with blanks, and the two text lines are put together to form the 120-character header line which titles the report about the data base field.

For Dictionary reports, the header text appears as entered, on two 60-character lines.

Processing Options

The options available for this form while in edit mode are:

1-PROC processes entered data
2-REGEN regenerates contents of current form
9-RTN returns to previously displayed FIELD form
10-HDR returns to HEADER form
11-EXPLAIN displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

JOB FORM

The JOB form is used to enter and display job definitions. The blank form in edit mode is shown in Figure 19.

Accessing the JOB Form

You can access this form from the HEADER form.

JOB	NAME:	OCC:	STAT:
DESC:			
=====			
ACT: _	1-PROC	2-REGEN	3-DESC
	6-USERDATA-NO: _	9-REUSE	10-HDR 11-EXPLAIN
(RESPONSE LINE)			

DISPLAY Mode:

=====			
ACT: _	2-REGEN	3-DESC	10-HDR 11-EXPLAIN
	6-USERDATA-NO: _	9-REUSE	
(RESPONSE LINE)			

DELETE Mode:

=====			
ACT: _	1-DELETE	9-REUSE	10-HDR 11-EXPLAIN
(RESPONSE LINE)			

Figure 19. JOB Form

Content

The NAME, OCC, and STAT fields at the top reflect the name, occurrence number, and status code of the subject of the job. (Subject code J=job is assumed for all job definitions.) The form may be used to enter one line of description text.

Working with the Form

If you modify the values specified in the NAME, OCC, and STAT fields at the top of the form, you are expected to select the REUSE action. In this event, changes to other parts of the screen will be lost.

(Note that a job definition cannot have an alias.)

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
3-DESC	displays DESCRIPTION form for subject
6-USERDATA_NO	selects set of User Data for subject
9-REUSE	redispays form for use with another subject
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

LCHILD FORM

The LCHILD form is used to enter and display relationship data about a logical child relationship between two segments in the same or different data bases. The blank form in edit mode is shown in Figure 20.

Accessing the LCHILD Form

You can access this form from the PHYSICAL SEGMENT form or from the LCHILD form for another logical child segment of the subject (logical parent) segment.

```

LCHILD
=====
(LOG-PARENT)-
  SEGMENT:      CODE:  OCC:  STAT:
  DATABASE:     STAT:

(LOG-CHILD)-
  SEGMENT:      CODE= A  OCC:  STAT:
  DATABASE:     CODE:    STAT:

          PTR:      PLACE:  SECINDX:

  PAIRNAME:     CODE= A  OCC:  STAT:
=====
ACT:  _  1-PROC  2-REGEN  3-DELETE          9-RTN  10-HDR  11-EXPLAIN
      8-LCHILD SEGMENT:  _____ CODE= A OCC:  _  STAT:  _
                                (RESPONSE LINE)
  
```

DISPLAY Mode:

```

=====
ACT:  _  2-REGEN          9-RTN  10-HDR  11-EXPLAIN
      8-LCHILD SEGMENT:  _____ CODE= A OCC:  _  STAT:  _
                                (RESPONSE LINE)
  
```

Figure 20. LCHILD Form

Content

The (LOG-PARENT) fields contain the subject (SEGMENT, CODE, OCC, and STAT) for the logical parent segment and the data base in which it occurs (DATABASE and STAT, CODE=P and OCC=000 are assumed). These names cannot be changed on this form.

(LOG-CHILD)

The fields SEGMENT, CODE, OCC, and STAT specify the complete Dictionary subject for the logical child segment. See Figure 8 for the subject rules.

The fields DATABASE, CODE, and STAT specify the Dictionary subject for the data base in which this LCHILD occurs. OCC=0 is assumed.

PTR

This field specifies the pointer. Valid values are:

S	Single
D	Double
I	Index
Y	Symbolic
N	None

PLACE

This field specifies the method used for inserting new segments in the data base. Valid values are:

F	First
L	Last
H	Here

SECINDX

This field specifies whether this is a secondary index relationship. A value of X specifies a secondary index relationship. If omitted, it is not a secondary index relationship.

The PAIRNAME, CODE, OCC, and STAT fields specify the complete Dictionary subject of a logical child segment logically or virtually paired with the logical child segment named above.

Working with the Form

The subject fields for (LOG-CHILD) SEGMENT, DATABASE, and PAIRNAME are treated as groups. If subject qualifiers are omitted, the established default qualifiers are assumed. The values in these fields can be modified at your discretion. Any modifications are treated as changes from one specified value to another, and not as changes in the subject being displayed.

Note that it is the logical child segment that serves to establish the LCHILD specification. The LOG-CHILD subject must have been defined before the form is submitted for processing (by selecting PROC).

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
3-DELETE	deletes subject; returns to previous form
8-LCHILD	displays LCHILD form for another subject
9-RTN	returns to previously displayed LOG-SEGMENT form
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

Specialized Actions

LCHILD: This action displays the LCHILD (logical child) form. When you choose this action, you should enter the complete segment name in the adjacent fields, unless you want to accept the current defaults for CODE, OCC, and STAT. If the segment you entered does not have a relationship already established, the subject name will be displayed on the LCHILD form without highlighting.

LOG-SEGMENT FORM

The LOG-SEGMENT form is used to enter and display the definition for a segment in a logical data base. The blank form in edit mode is shown in Figure 21.

Accessing the LOG-SEGMENT Form

You can access this form from the HEADER form (if the segment was given the role of a logical segment), by an automatic transfer from the no-role SEGMENT form once the role of this segment as a logical segment is established, or by specifying the REUSE action on another SEGMENT form.

Contents

The fields at the top of the LOG-SEGMENT form are treated as outlined under "Common Fields in SEGMENT Forms" in the "No-role SEGMENT Form" section below. Note that the FREQ parameter is not applicable to a logical segment.

The (SOURCE1) and (SOURCE2) fields are used to identify the first and second source segments for the logical segment (that is, the physical segments to which the logical segment corresponds) and to specify whether or not the data portion of either of the source segments is to appear in the logical segment. Note that the segments you specify are assumed to be defined in the Dictionary; the Dictionary does not check these entries.

Each (SOURCE) specification has seven labeled fields.

The SEGMENT, CODE, OCC, and STAT fields specify the complete Dictionary subject name for the source segment.

KEY/DATA

This field specifies whether the logical segment contains only the key of the source segment or the actual data. Valid values are:

K	key only
D	data

The DATABASE and STAT fields specify the name and status of the data base to which the source segment is related. Note that in the SOURCE= parameter of a SEGM statement for DBD generation, the KEY/DATA parameters would be specified between the source

LOG-SEGMENT	NAME:	OCC:	STAT:
	CODE:		
DESC:			
DATE:	(MAX)BYTES:	MINBYTES:	NOFLDS: ALIGNED:
DATABASE:	CODE: L	STAT:	ROLE= L
(SOURCE1)-			
SEGMENT:	CODE= A	OCC:	STAT:
KEY/DATA:			
DATABASE:	STAT:		
(SOURCE2)-			
SEGMENT:	CODE= A	OCC:	STAT:
KEY/DATA:			
DATABASE:	STAT:		
=====			
ACT: _	1-PROC	2-REGEN	3-DESC 4-ALIAS 5-FLDS 7-PL/I 10-HDR 11-EXPLAIN
	6-USERDATA-NO: _	9-REUSE	
(RESPONSE LINE)			

DISPLAY Mode:

=====										
ACT: _		2-REGEN	3-DESC	4-ALIAS	5-FLDS	7-PL/I	10-HDR	11-EXPLAIN		
	6-USERDATA-NO: _	9-REUSE								
(RESPONSE LINE)										

Figure 21. LOG-SEGMENT Form

segment and data base names.

Working with the Form

The segment and data base subject fields in the (SOURCE) specifications are treated as groups. If OCC or STAT qualifiers are omitted, the current default qualifiers are assumed (note, however, that CODE=A is required for the source segments; CODE=P and OCC=0 are assumed for the source data base). Modifications to any of these fields are treated as changes from one specified data base or segment to another and not as changes in the names of the data base or segment now displayed.

The values in the KEY/DATA fields can be modified at your discretion.

If you modify the values specified in the NAME, OCC, and STAT fields at the top of the form or in the DATABASE line, you are expected to select the REUSE action. In this event, changes to other parts of the screen will be lost.

If you select REUSE, the form you get will depend upon whether the segment you specify has a relationship established with the data base displayed in the DATABASE field or not. If it does, the role-dependent form appropriate for that relationship will be displayed. If no relationship has been established, the no-role SEGMENT form will be displayed. You may then select the appropriate role-dependent form from the no-role SEGMENT form.

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
3-DESC	displays DESCRIPTION form for subject
4-ALIAS	displays ALIASES form for subject
5-FLDS	displays SEGMENT-FIELDS form
6-USERDATA_NO	selects set of User Data for subject
7-PL/I	displays PL/I DATA form for segment or field
9-REUSE	displays appropriate role-dependent form
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field or select a PF key.

Specialized Actions

PL/I: This action displays the PL/I DATA form associated with the particular segment or field defined on the form from which this action is selected.

LOGICAL DATABASE FORM

The LOGICAL DATABASE form is the key form for defining a DL/I logical data base. A logical data base interrelates segments of one or more physical data bases in a hierarchical order corresponding to their use in an application. The blank form in edit mode is shown in Figure 22.

Accessing the LOGICAL DATABASE Form

You can access this form from the HEADER form.

Content

The NAME and STAT fields at the top show the subject for the data base. CODE=L, for logical, and OCC=000 are assumed. See Figure 8 for subject rules.

PASSWD

(VSAM only.) This field specifies whether the DBDNAME is used as the VSAM password. Valid values are Y for yes and N for no.

SECURITY

QUERY

This field specifies an access code for read-only access of a field in a data base. Any three alphameric characters are valid.

If GIS data base security is enabled, values may be numeric only, from 0 to 128. The value 0 means there is no security access code required.

UPDATE

This field specifies an access code for authorization to update a field in a data base. Any three alphameric characters are valid.

If GIS data base security is enabled, values may be numeric only, from 0 to 128. The value 0 means no security access code is required.

LOGICAL DATABASE	NAME:	STAT:		
DESC:				
PASSWD:				
SECURITY - QUERY:	UPDATE:	DATMUSER-NO:	DATMUSER-SEQ:	
<pre> ===== ACT: _ 1-PROC 2-REGEN 3-DESC 4-ALIAS 5-STRUC 10-HDR 11-EXPLAIN 6-USERDATA-NO: _ 9-REUSE (RESPONSE LINE) ===== </pre>				

DISPLAY Mode:

<pre> ===== ACT: _ 2-REGEN 3-DESC 4-ALIAS 5-STRUC 10-HDR 11-EXPLAIN 6-USERDATA-NO: _ 9-REUSE (RESPONSE LINE) ===== </pre>				
--	--	--	--	--

DELETE Mode:

<pre> ===== ACT: _ 1-DELETE 9-REUSE 10-HDR 11-EXPLAIN (RESPONSE LINE) ===== </pre>				
---	--	--	--	--

Figure 22. LOGICAL DATABASE Form

DATMUSER-NO

This field specifies which User Data set (1 to 5) is to hold data set specifications for the subject data base.

DATMUSER-SEQ

This field specifies the starting sequence number in the User Data set specified in the DATMUSER-NO field. (The range of available User Data lines is 1 to 999.) For GIS, the content of the User Data could be JCL statements or DATM statements.

Working with the Form

If you modify the values specified in the NAME, OCC, and STAT fields at the top of the form, you are expected to select the REUSE action. In this event, changes to other parts of the screen will be lost.

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
3-DESC	displays DESCRIPTION form for subject
4-ALIAS	displays ALIASES form for subject
5-STRUC	displays LOGICAL DATABASE STRUCTURE form
6-USERDATA_NO	selects set of User Data for subject
9-REUSE	redispays form for use with another subject
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

Specialized Actions

STRUC: This action displays the appropriate STRUCTURE form.

LOGICAL DATABASE STRUCTURE FORM

The LOGICAL DATABASE STRUCTURE form is a window form that is used to enter and display the hierarchical structure of the segments of the subject logical data base. The segments are listed in hierarchical order, with their level and parent segments indicated. A single logical data base can have as many as 255 segment types in its hierarchical structure. The blank form in edit mode is shown in Figure 23.

Accessing the LOGICAL DATABASE STRUCTURE Form

You can access the LOGICAL DATABASE STRUCTURE form from the LOGICAL DATABASE form.

Content

The DATABASE and STAT fields at the top give the user name and status for the subject data base. CODE=L, for logical, and OCC=000 are assumed. The data base subject cannot be modified on this form. The table has columns for the following entries:

The SEGMENT, CODE, OCC, and STAT fields specify the complete Dictionary subject for this segment.

LEVL

This field specifies the level of this segment in the data base hierarchy. The value may be 1 to 15.

The PARENT, CODE, OCC, and STAT fields specify the complete Dictionary subject for the parent segment.

MODE

This field specifies whether the segment definition is new or previously defined. The value can be:

N New (the default)

O Old

* Move with dependents (see note at the end of "Working with the Form")

The last line of the table is used for INSERT and MOVE actions.

LOGICAL DATABASE STRUCTURE				DATABASE:		STAT:			
SEGMENT	CODE	OCC	STAT	LEVL	PARENT	CODE	OCC	STAT	MODE
===== ACT: _ 1-PROC 2-REGEN 3-TOP 4-INSERT 5-MOVE 9-RTN 10-HDR 11-EXPLAIN 6-DOWN: 12_ POS BEFORE SEGMENT: _____ CODE: A OCC: _ STAT: _ (RESPONSE LINE)									

DISPLAY Mode:

===== ACT: _ 2-REGEN 3-TOP 9-RTN 10-HDR 11-EXPLAIN 6-DOWN: 12_ (RESPONSE LINE)									
--	--	--	--	--	--	--	--	--	--

Figure 23. LOGICAL DATABASE STRUCTURE Form

Working with the Form

When the form is first presented, the window is positioned at the top of the list of segments related to this data base. You can move the window down over the list with DOWN action. To return to the beginning of the list, select TOP.

To enter new relationships between segments and this data base, first position the window so that one or more empty lines appear at the bottom of the table. Then key in the segment specifications, enter N or O in the MODE column (or accept the default N), and select PROC for processing. Only the segment subject is required to establish a relationship; if you omit any qualifiers, the current defaults are assumed. If your entry is not in correct hierarchical order, it can be moved with the MOVE action.

If the segment you specify has not been previously defined in the Dictionary, an entry in this table will establish a Dictionary definition for that segment as well as its relationship to the subject data base. (If you enter N for "New" or accept the N default as the MODE column entry, and there is already a segment with this subject defined in the Dictionary, you will receive an error message.)

If you are establishing a relationship between the subject data base and a segment previously defined, you enter O for "Old" in the MODE column. The Dictionary checks to see that there is such a segment definition. If the segment is not found, you will receive an error message. If the segment is found, the relationship will be established when your entries are processed.

You are required to enter DL/I names (subjects having CODE=A and a user name eight or fewer characters in length) for the segments on this form. If you enter a segment subject with a CODE value other than A, you will receive an error message. Further, if in retrieving the list of segment names to be displayed on the form one (or more) of the related segments is found not to have a DL/I alias, the form will be displayed with an appropriate error message, and the mode of processing will be automatically switched to display mode.

Once entered in the table, a segment subject cannot be modified with this form.

On this form you may modify the LEVL and PARENT subjects, which are considered parameters of the segment-data base relationship (that is, relationship data). The PARENT parameters must be treated as a group; if you omit any qualifiers, the current defaults are assumed.

The MODE column is processed only for new entries in the table, and not for changed or added parameters.

To delete an entry from the list, erase the SEGMENT (user name) portion of the subject with minus signs. On processing, the relationship between the specified segment and the subject data base is deleted from the Dictionary. The definitions stored in the Dictionary for the segment and the data base are not affected. Any or all of the segment entries displayed on the form at a given time can be deleted in this way. To insert a segment name within the list, use the line set off at the bottom of the table. Enter the segment subject and any other parameters you want to specify, indicate after POS BEFORE SEGMENT the subject of the segment before which the insert is to be made, and select INSERT. If you omit any qualifiers, the current defaults are assumed.

To move a segment already listed to another position within the list, use the same line. Enter the subject in the line, and new values for LEVEL and PARENT (if desired). Indicate after POS BEFORE SEGMENT the subject of the segment before which this segment is to be inserted, and select MOVE. Neither the present nor the destination position of the segment has to be displayed in the window when the INSERT or MOVE action is performed.

If you enter an asterisk (*) in the MODE column before selecting the MOVE action, a RELOCATE WITH DEPENDENTS command will be issued for the specified segment (see "RELOCATE Command" in Chapter 6).

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
3-TOP	positions window at top of stored list
4-INSERT	inserts a segment name into table
5-MOVE	moves an existing segment into table
6-DOWN	moves window down over stored list
9-RTN	returns to previously displayed LOGICAL DATABASE form
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

Specialized Actions

INSERT: This action causes a subject name, typed in a special display line just above the action area, to be inserted in the stored list for this form, at the position you specify following the POS BEFORE SEGMENT field on the next line. The segment that the inserted segment is to precede need not appear on the

display.

MOVE: This action moves a segment name already present in a stored list to the position specified by your entry in the POS BEFORE SEGMENT field on the next line. Neither the segment to be moved nor the segment before which it is to be moved has to appear on the display.

POS BEFORE SEGMENT: These fields in the action area do not have a number associated with them. They are always used with either the INSERT or MOVE action. The values you supply in these fields specify the name of the segment before which a segment is to be moved or inserted.

MODULE FORM

The MODULE form is used to enter and display module definitions. The blank form in edit mode is shown in Figure 24.

MODULE	NAME:	OCC:	STAT:		
DESC:					
=====					
ACT: _	1-PROC	2-REGEN	3-DESC	10-HDR	11-EXPLAIN
	6-USERDATA-NO: _	9-REUSE	(RESPONSE LINE)		

DISPLAY Mode:

=====					
ACT: _	2-REGEN	3-DESC	10-HDR	11-EXPLAIN	
	6-USERDATA-NO: _	9-REUSE	(RESPONSE LINE)		

DELETE Mode:

=====					
ACT: _	1-DELETE	9-REUSE	10-HDR	11-EXPLAIN	
		(RESPONSE LINE)			

Figure 24. MODULE Form

Accessing the MODULE Form

You can access this form from the HEADER form.

Content

The NAME, OCC, and STAT fields at the top of the form give the name, occurrence number, and status code of the subject for this module. Subject code M=module is assumed. The remainder of the form may be used to enter one line of description text.

Working with the Form

If you modify the values specified in the NAME, OCC, and STAT fields at the top of the form, you are expected to select the REUSE action. In this event, changes to the screen below these fields will be lost. (Note that a module cannot have aliases.)

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
3-DESC	displays DESCRIPTION form for subject
6-USERDATA_NO	selects set of User Data for subject
9-REUSE	redispays form for use with another subject
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

PCB FORM

The PCB form is the key form for entering and displaying definitions of PCBs. The form can accept information about data base (DB), alternative output (TP), and GSAM PCBs. The blank form in edit mode is shown in Figure 25.

Accessing the PCB Form

You can access the PCB form from the HEADER form.

Content

The (PSB)-NAME, OCC(SEQ), and STAT fields at the top of the form show the subject for the PCB (CODE=P is assumed). The PCBs that occur within a given PSB (Program Specification Block) are usually identified by the same name and status code as that PSB. The occurrence number of each PCB is then used to indicate its position in the PSB. The form has additional labeled fields for entries as follows:

TYPE

This field specifies the type of PCB. Valid characters are:

D	Data base
T	Telecommunication
G	GSAM

(TYPE=TP)

The following fields specify the parameters for a telecommunication PCB.

PCB	(PSB)-NAME:	OCC(SEQ):		
	STAT:			
DESC:				
TYPE:				
(TYPE=TP)-	LTERM/TRANS:	MODIFY:	EXPRESS:	
	SAMETERM:	ALTRESP:		
(TYPE=DB)-	DATABASE:	CODE:	STAT:	
	PROCSEQ:	PROCOPT:	KEYLEN:	POS:
	SECURITY - QUERY	UPDATE:		
(TYPE=GSAM)-	DATABASE:	CODE:	STAT:	
		PROCOPT:		
=====				
ACT: _	1-PROC	2-REGEN	3-DESC	4-ALIAS 5-DB-SENSEGS 10-HDR 11-EXPLAIN
	6-USERDATA-NO: _	9-REUSE		
(RESPONSE LINE)				

DISPLAY Mode:

=====				
ACT: _	2-REGEN	3-DESC	4-ALIAS	5-DB-SENSEGS 10-HDR 11-EXPLAIN
	6-USERDATA-NO: _	9-REUSE		
(RESPONSE LINE)				

DELETE Mode:

=====				
ACT: _	1-DELETE			10-HDR 11-EXPLAIN
		9-REUSE		
(RESPONSE LINE)				

Figure 25. PCB Form

LTERM/TRANS

This field specifies the name of the logical terminal or transaction code that is the destination of all output messages. The value can be up to 8 alphameric characters.

MODIFY

This field specifies whether the PCB destination may be modified. Valid characters are:

- Y Yes
- N No

EXPRESS

This field specifies whether messages should be sent if a program abends. Valid characters are:

- Y Yes, send
- N No, back out

SAMETERM

This field specifies whether IMS/VS should check that the PCB's logical terminal is assigned to the same physical

terminal as the I/O PCB. A value of Y (yes) specifies that IMS/VS should check. If no value is specified, IMS/VS will not check.

ALTRESP

This field specifies whether IMS/VS should accept messages to this PCB as conversational responses. Valid characters are:

Y Yes

N No

(TYPE=DB)

The following fields specify the parameters for PCBs with TYPE=D (data base).

The DATABASE, CODE, and STAT fields specify the Dictionary subject of the target data base for the PCB. The value OCC=0 is assumed.

The next four fields--PROCSEQ, PROCOPT, KEYLEN, and POS--correspond to those specified on a TYPE=DB PCB statement when generating a PSB:

PROCSEQ

This field specifies the name of the secondary index data base used to define the PCB processing sequence. The value can be up to 8 alphanumeric characters.

PROCOPT

This field specifies the processing options to be used with sensitive segments declared in the PCB. Valid values are:

G Get

I Insert

R Replace

D Delete

A All the above

P Command code used in GET

E Exclusive use

L Load (except for HIDAM)

S Segments in ascending sequence only

GO Get (without program isolation)

GS Get segments in ascending sequence only (HSAM only)

LS Segments loaded in ascending sequence only (HIDAM, HDAM)

KEYLEN

This field specifies the length in bytes of the longest concatenated key in the sensitive segment path. The value may be 1 to 3825.

POS

This field specifies whether single or multiple positioning is to be used in accessing the logical data structure. Valid values are:

S Single

M Multiple

SECURITY

QUERY

This field specifies an access code for read-only access of a field of a data base. Any three alphameric characters are valid.

If GIS data base security is enabled, values can be numeric only, from 0 to 128. The value 0 means there is no security access code required.

UPDATE

This field specifies an access code to be used in updating a data base. Any three alphameric characters are valid.

If GIS data base security is enabled, values must be numeric, from 0 to 128. The value 0 means no security access code is required.

(TYPE=GSAM)

The following fields specify the parameters for PCBs with TYPE=G.

The DATABASE, CODE, and STAT fields specify the Dictionary subject of the target data base for the PCB. The value OCC=0 is assumed.

PROCOPT

This field specifies the processing options to be used with sensitive segments declared in this PCB. Valid values are:

G	Get
L	Load
S	Large-scale sequential activity

Working with the Form

If you modify the values specified in the NAME, OCC, and STAT fields at the top of the form, you are expected to select the REUSE action. In this event, changes to other parts of the screen will be lost.

The values in any of the (TYPE=TP) fields can be modified at your discretion.

In the (TYPE=DB) specifications, the three data base subject fields must be treated as a group. If any subject qualifier is omitted, the current defaults are assumed. An entry of a data base subject here causes a relationship to be established between the data base definition and the definition for the subject PCB. If you enter a data base subject that is not already stored in the Dictionary, an entry will be added to the Dictionary. Once specified, the data base subject cannot be modified on this form. You may, however, delete the relationship between the definitions for the subject PCB and the specified data base by simply erasing (writing over with minus signs) the DATABASE field and selecting PROC. The relationship is then deleted, but the definitions of the PCB and data base are not affected.

The PROCSEQ, PROCOPT, KEYLEN, and POS fields may be modified at your discretion.

The (TYPE=GSAM) fields specify parameters for a PCB with TYPE=GSAM. The three DATABASE subject fields and the PROCOPT field are treated in the same way as the corresponding (TYPE=DB) parameters.

Accessing the PCB SENSEGS Form

You can access the PCB SENSEGS form from the associated PCB form.

Content

The (PSB)-NAME, OCC(SEQ) and STAT fields at the top give the subject name for the PCB (CODE=P is assumed). The subject cannot be modified on this form.

The table in the center of the form is the window on the stored list of sensitive segments for the PCB. The table has 13 columns. The first four are for the subject of the sensitive segments listed (SEGMENT, CODE, OCC, and STAT). The next four (PARENT, CODE, OCC, and STAT) name the parent segment within the sensitive segment. The remaining five columns have headings as follows:

PROC

This field specifies the processing options that may be used with the PCB's sensitive segments. Valid characters are:

G	Get
I	Insert
R	Replace
D	Delete
A	All of the above
P	Command code
E	Exclusive use
L	Load function (except for HIDAM)
S	Segments in ascending sequence only
GS	Get segments in ascending sequence only (HSAM only)
LS	Segments loaded in ascending sequence only (HIDAM, HDAM)
K	Key sensitivity only

INDEX1 INDEX2 INDEX3

These fields specify the names of the first three secondary indexes that contain search fields qualifying SSAs for this sensitive segment. The value may be 1 to 8 alphameric characters.

M

This field indicates the sensitive segment has secondary indexes defined in the Dictionary with numbers other than 1, 2, or 3. Enter a ? over the M if you wish to see the other secondary indexes. Entering * indicates move with dependents (see note at end of the "Working with the Form" section). The bottom line of the table, just above the action area, is used for INSERT and MOVE actions, as explained below.

Working with the Form

When first presented, the form is positioned at the top of the list of sensitive segments for the PCB. You can use the DOWN action to move the window down over the list of segments. You can return to the beginning of the list by selecting TOP.

To create new entries in the list (that is, to define a relationship between this PCB and segments not previously listed), first position the window so that one or more empty lines appear at the bottom of the table. Then key the new sensitive segment specification into these lines. Only the segment subject name (SEGMENT, CODE, OCC, and STAT) has to be specified to establish the relationship. If you omit any subject qualifiers, the current defaults are assumed. Select PROC to store your entries in the Dictionary.

You are required to enter DL/I names (subjects having CODE=A and a user name eight or fewer characters in length) for the segments on this form. In retrieving the list of sensitive segment names to be displayed on the form, if one (or more) of the sensitive segments is found not to have a DL/I alias, the form will be displayed with an error message, and the mode of processing will be automatically switched to display mode.

Processing of new entries on this form does not automatically create new definitions for segments. If you enter a segment subject that is not already defined in the Dictionary, you will receive an error message.

Once entered in the table, a sensitive segment subject cannot be modified on this form.

An entry can be deleted from the list by erasing (overlying with minus signs) the user name (SEGMENT column entry) of the segment. On processing, the relationship between the specified segment and the subject PCB is deleted from the Dictionary and disappears from the display form. The stored definitions of the segment and the PCB are unaffected. Any or all of the segment entries displayed on the form at a given time can be deleted in this way.

The PARENT subject fields and INDEXES fields are considered parameters of the segment-PCB relationship and may be modified at your discretion. The PARENT subject fields must be treated as a group; if you omit any qualifiers, the current defaults are assumed.

A single DL/I sensitive segment may have up to 32 secondary indexes. In the Dictionary, these indexes are numbered in sequence. The INDEX columns of the table specify the names of the first three indexes. If an index with a sequence number greater than 3 has been defined in the Dictionary for a given sensitive segment, an M is displayed in the M column following the entry for that segment. If you want to review, add to, or edit the complete list of secondary indexes for the segment, enter a question mark ? in the M column in the line for that segment. (Enter the question mark over the M if one is displayed.) Then select the MORE-IND action, and the sensitive segment indexes form, shown in Figure 27, for this segment will be displayed. Only one question mark may be entered in the M column before the MORE-IND action is selected. (That is, you can request the sensitive segment indexes for just one sensitive segment at a time.) A ? in the M column is only valid for the MORE-IND (PFK7) action.

The line at the bottom of the table, set off from the rest and just above the action area, is used to enter the subject and other parameters for a sensitive segment that you wish to insert or move within the list. To insert a sensitive segment, enter the segment subject (and any other parameters) in this line, fill in the POS BEFORE SEGMENT fields to specify the point in the list at which the new sensitive segment is to be inserted, and select INSERT. To move a sensitive segment within the list, enter the segment subject, a new value for PARENT (if desired),

fill in the POS BEFORE SEGMENT fields to specify the point in the list to which the sensitive segment is to be moved, and select MOVE.

Note: If you enter an asterisk (*) in the M column before selecting the MOVE action, a RELOCATE WITH DEPENDENTS command will be issued for the specified sensitive segment (see the "RELOCATE Command" in Chapter 6).

Neither the destination of the inserted or moved segment nor the segment being moved has to appear in the window currently displayed.

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
3-TOP	positions window at top of stored list
4-INSERT	inserts a segment into stored list
5-MOVE	moves segment in stored list
6-DOWN	moves window down over stored list
7-MORE-IND	displays PCB SENSEG INDEXES form
9-RTN	returns to previously displayed PCB form
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

Specialized Actions

INSERT: This action causes a subject name, typed in a special display line just above the action area, to be inserted in the stored list for this form, at the position you specify following the POS BEFORE SEGMENT field on the next line. The segment that the inserted segment is to precede need not appear on the display.

MOVE: This action moves a segment name already present in a stored list to the position specified by your entry in the POS BEFORE SEGMENT field on the next line. Neither the segment to be moved nor the segment before which it is to be moved has to appear on the display.

POS BEFORE SEGMENT: These fields in the action area do not have a number associated with them. They are always used with either the INSERT or MOVE action. The values you supply in these fields specify the name of the segment before which a segment is to be moved or inserted.

MORE-IND: This action displays the PCB SENSEG INDEXES form, which lists the indexed fields for a particular sensitive segment. When you select this action from the PCB SENSEGS form, you have to indicate which segment you are interested in by entering a question mark (?) in the M (for More) column in the appropriate segment row.

PCB SENSEG INDEXES FORM

The PCB SENSEG INDEXES form is used for specifications of secondary indexes when there are more than three indexes for a given sensitive segment.

A PCB sensitive segment can have up to 32 secondary indexes associated with it. The PCB SENSEGS form (See Figure 26) includes specifications for the first three secondary indexes for each sensitive segment listed. The blank PCB SENSEG INDEXES form in edit mode is shown in Figure 27

PCB SENSEG INDEXES				
(SENSEG)-	NAME:	CODE:	OCC:	STAT:
(PCB)-	(PSB)-NAME:	OCC(SEQ):	STAT:	
INDEX	INDEX	INDEX	INDEX	
01	09	17	25	
02	10	18	26	
03	11	19	27	
04	12	20	28	
05	13	21	29	
06	14	22	30	
07	15	23	31	
08	16	24	32	

ACT: _ 1-PROC 2-REGEN (RESPONSE LINE) 9-RTN 10-HDR 11-EXPLAIN				

DISPLAY Mode:

=====				
ACT: _	2-REGEN	(RESPONSE LINE)	9-RTN	10-HDR 11-EXPLAIN

Figure 27. PCB SENSEG INDEXES Form

Accessing the PCB SENSEG INDEXES Form

You can access the PCB SENSEG INDEXES form by selecting the MORE-IND action on the associated PCB SENSEGS form.

Content

The (SENSEG) fields show the subject for the sensitive segment of interest. This subject cannot be modified. Similarly, the (PCB) fields, which give the subject of the PCB to which this segment is related, cannot be modified. The table at the bottom of the form is used to specify the names of the 1 to 32 secondary indexes that are to be used in conjunction with the subject sensitive segment. Note that the first three entries in the table (sequence numbers 01 to 03) correspond to the values specified in the INDEX1, INDEX2, and INDEX3 columns in the entry for the sensitive segment on the PCB SENSEGS form.

Working with the Form

The index names specified in the table can be modified at your discretion. If you erase an index name (by writing over it with minus signs), that name will be dropped from the list of indexes associated with the subject segment. (The Dictionary definitions of the secondary index and the segment are unaffected.) The sequence number of an erased entry is left unused and can be reused to specify another index name.

Processing Options

The options available for this form while in edit mode are:

- 1-PROC processes entered data
- 2-REGEN regenerates contents of current form
- 9-RTN returns to previously displayed PCB SENSEGS form
- 10-HDR returns to HEADER form
- 11-EXPLAIN displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

PHYSICAL DATABASE FORM

The PHYSICAL DATABASE form is the key display form for defining a DL/I physical data base. You can use this form to enter and display information about the data base, the DL/I and underlying OS access methods used with the data base, and the data set groups in which the data base is to be stored. The blank form in edit mode is shown in Figure 28.

PHYSICAL DATABASE		NAME:	STAT:										
DESC:													
DBACCS:	(HDAM) -	RMNAME:	MAXRBN:										
OSACCS:		ANCH:	MAXLEN:										
PASSWD:	(GSAM) -	RECFM:											
SECURITY - QUERY:		UPDATE:	DATMUSER-NO:	DATMUSER-SEQ:									
DSG	LABEL	DD1	BLK1	RECL1	DD2/OVFLW	BLK2	RECL2	DEV	MOD	SCAN	FSPF	FBFF	
01													
02													
03													
04													
05													
06													
07													
08													
09													
10													
=====													
ACT: _		1-PROC	2-REGEN	3-DESC	4-ALIAS	5-STRUC					10-HDR	11-EXPLAIN	
		6-USERDATA-NO: _	9-REUSE										
(RESPONSE LINE)													

DISPLAY Mode:

=====												
ACT: _		2-REGEN	3-DESC	4-ALIAS	5-STRUC						10-HDR	11-EXPLAIN
		6-USERDATA-NO: _	9-REUSE									
(RESPONSE LINE)												

DELETE Mode:

=====												
ACT: _		1-DELETE									10-HDR	11-EXPLAIN
			9-REUSE									
(RESPONSE LINE)												

Figure 28. PHYSICAL DATABASE Form

Accessing the PHYSICAL DATABASE Form

You can access the PHYSICAL DATABASE form from the HEADER form.

Content

The NAME and STAT fields at the top give the subject name of the data base. CODE=P (for a physical data base) and OCC=0 are assumed. The form has labeled fields for the following entries:

DBACCS

This field specifies the DL/I access method used with this data base. The value can be HDAM, HSAM, SHSAM, SHISAM, HIDAM, GSAM, or HISAM.

OSACCS

This field specifies the OS/VS access method. The value can be BSAM, OSAM, VSAM, or ISAM.

PASSWD

(VSAM only.) This field specifies whether the DBNAME is used as the VSAM password. The value is Y for Yes or N for No.

(HDAM)

RMNAME

This field specifies the randomizing module name. It is required for HDAM. The value can be 1 to 8 alphameric characters.

ANCH

This field specifies the number of anchor points for each block. It is required for HDAM. The value can be 0 to 255.

MAXRBN

This field specifies the value of the maximum OSAM relative block number for the randomizing module. This field is used only with HDAM. The value can be 0 to 1677215.

MAXLEN

This field specifies the maximum length (in bytes) of an insert in the data base root segment in a single insert sequence. It is used only with HDAM. The value can be 0 to 1677215.

(GSAM)

RECFM

This field specifies the record format for OS/VS data sets and GSAM data bases. See the OS/VS JCL Reference for the valid values.

SECURITY

QUERY

This field specifies an access code for read-only access of a field of a data base. Any three alphameric characters are valid.

If GIS data base security is enabled, values can be numeric only, from 0 to 128. The value 0 means there is no security access code required.

UPDATE

This field specifies an access code to be used in updating a data base. Any three alphameric characters are valid.

If GIS data base security is enabled, values must be numeric, from 0 to 128. The value 0 means no security access code is required.

DATMUSER-NO

This field specifies which User Data set (1 to 5) is to hold data set specifications for the subject data base.

DATMUSER-SEQ

This field specifies the starting sequence number in the User Data set specified in the DATMUSER-NO field. (The range of available User Data lines is 1 to 999.) For GIS, the content of the User Data set could be JCL statements or DATM statements.

Below these labeled fields is a table for specifying the data set groups (DSGs) in which the data base is to be stored. In the Dictionary, each DSG is identified by a DSG-number from 01 to 10 (first column). The remaining columns are used to specify parameters corresponding to those in the DATASET statement for DL/I DBDGEN. (DD1 is the primary data set; DD2 is a secondary data set for overflow or some other purpose.) Entries in these columns are as follows:

LABEL

This field specifies the identifier for the data set group. The value may be 1 to 8 alphameric characters.

DD1

This field specifies the ddname of the primary data set for this data set group. It is required for DBD_OUT. The value may be 1 to 8 alphameric characters.

BLK1

This field specifies the block size for DD1. It is optional for DBD_OUT. The value may be 0 to 32767.

RECL1

This field specifies the logical record length for DD1. It is optional for DBD_OUT. The value may be 0 to 32767.

DD2/OVFLW

This field specifies the ddname of the secondary/overflow data set. It is optional for DBD_OUT. The value can be 1 to 8 alphameric characters.

BLK2

This field specifies the block size for DD2. It is optional for DBD_OUT. The value can be 0 to 32767.

RECL2

This field specifies the logical record length for DD2. It is optional for DBD_OUT. The value can be 0 to 32767.

DEV

This field specifies the type of physical storage device for this DSG. It is required for DBD_OUT. Typical values are: 2319, 2314, 2305, 3330, 3340, 3350, 3400, 2400, or TAPE.

MOD

This field specifies the model of 2305 or 3330. The value can be 1 or 2 digits with the following meaning:

1	Model of 2305 or 3330
2	Model of 2305
11	Model of 3330

SCAN

This field specifies the number of DASD cylinders that will be scanned in search for space to insert segments into an HDAM or HIDAM data base. It is optional for DBD_OUT. A maximum of three digits with a value of 0 to 255 may be specified. If omitted, the default is 3.

FSPC

This field specifies the percentage of free space in a HDAM or a HIDAM data base. The value may be 0 to 99.

FBFF

This field specifies the free block frequency factor for a HDAM or a HIDAM data base. The value may be 0 to 100, but not 1. If the value specified for DBACCS is HSAM, SHSAM or GSAM, then the values entered or displayed in the DD2/OVFLW column are assumed to be DD2 values; otherwise, the values in the DD2/OVFLW column are assumed to be OVFLW values.

Working with the Form

If you modify the values specified in the NAME, OCC, and STAT fields at the top of the form, you are expected to select the REUSE action. In this event, changes to other parts of the screen will be lost.

The values specified in the DBACCS, OSACCS, GSAM, PASSWD, SECURITY, and HDAM fields may be modified at your discretion.

You can store a DSG specification by entering values into one or more fields in the appropriate row of the table. All fields of the DSG specification need not be complete for processing; some are applicable only for particular data base access methods. All values in a DSG specification can be modified directly.

You may delete an entire DSG specification by erasing the DSG number and selecting PROC. The corresponding entry is removed from the table, and you may enter a new specification in the empty line. (The DSG number is redisplayed after processing.)

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
3-DESC	displays DESCRIPTION form for subject
4-ALIAS	displays ALIASES form for subject
5-STRUC	displays PHYSICAL DATABASE STRUCTURE form
6-USERDATA_NO	selects set of User Data for subject
9-REUSE	redispays form for use with another subject
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

STRUC: This action displays the appropriate STRUCTURE form.

PHYSICAL DATABASE STRUCTURE FORM

The PHYSICAL DATABASE STRUCTURE form is used to enter or display a list of the segments occurring in a data base. The order of the segments in the stored list indicates their hierarchical order in the data base. The form serves as a window (with a maximum of 12 lines) that can be moved up and down over the list of segments related to the data base, with their segment level and PARENT parameters.

Though the Dictionary will allow more, DL/I restricts a single physical data base to 255 segment types. The segment types may appear at one of 15 different levels in the hierarchical structure of the data base. The blank form in edit mode is shown in Figure 29

PHYSICAL DATABASE STRUCTURE				DATABASE:		STAT:					
SEGMENT	CODE	OCC	STAT	LEVL	PARENT	CODE	OCC	STAT	ROLE	DSG	MODE
===== ACT: _ 1-PROC 2-REGEN 3-TOP 4-INSERT 5-MOVE 9-RTN 10-HDR 11-EXPLAIN 6-DOWN: 12_ POS BEFORE SEGMENT: _____ CODE: A OCC: _ STAT: _ (RESPONSE LINE)											

DISPLAY Mode:

===== ACT: _ 2-REGEN 3-TOP 9-RTN 10-HDR 11-EXPLAIN 6-DOWN: 12_ (RESPONSE LINE)											
--	--	--	--	--	--	--	--	--	--	--	--

Figure 29. PHYSICAL DATABASE STRUCTURE Form

Accessing the PHYSICAL DATABASE STRUCTURE Form

You can access the PHYSICAL DATABASE STRUCTURE form from the PHYSICAL DATABASE form.

Content

The DATABASE and STAT fields at the top give the subject of the data base. (CODE=P and OCC=000 are assumed.) The subject cannot be modified on this form.

The table has columns for the following entries:

The SEGMENT, CODE, OCC, and STAT fields specify the complete Dictionary subject for this segment.

LEVL

This field specifies the level of this segment in the data base hierarchy. The value may be 1 to 15.

The PARENT, CODE, OCC, and STAT fields specify the complete Dictionary subject for the physical parent of this segment.

ROLE

This field specifies the role the subject segment plays in the data base. Valid values are:

- P Physical
- V Virtual logical child

DSG

This field specifies the number of the data set group in which occurrences of the segment are to be physically stored. The value can be 1 to 10.

MODE

This field specifies whether the segment definition is new or was previously defined. The value may be:

N	New
0	Old
*	Move with dependents (see note at the end of "Working with the Form" section below.)

If omitted, the default is N.

The last line of the table is used for INSERT and MOVE actions, as explained below.

Working with the Form

When the form is first presented, the window is positioned at the top of the list of segments related to this data base. You can move the window down over the list by selecting DOWN. To return to the beginning of the list, select TOP.

To enter new segment relationships, position the window so that one or more empty lines appear at the bottom of the table. Then key in the segment specifications and select PROC. Only the segment subject is required to establish a relationship; if you omit any qualifiers, the current defaults are assumed. The order of segments in the list represents their hierarchical order in the data base. (You can change the order with the MOVE or INSERT actions described below.)

If a segment you specify has not been previously defined, entry of the segment name in this table will establish a definition for that segment. (If you enter N for New or omit the MODE column entry and the Dictionary finds that there is already a segment with this subject defined in the Dictionary, you will receive an error message after your entries are processed.)

To establish a relationship between the subject data base and a segment previously defined, key in the segment subject (again, default qualifiers apply) and enter 0 in the MODE column. When you select PROC (after one or more entries), the Dictionary checks to see if there is such a segment definition. If the segment is not found, you will receive an error message. If the segment is found, the relationship is established on processing of your entries.

You are required to enter DL/I names (subjects having a code of A and a user name eight or fewer characters in length) for the segments on this form. If you enter a segment subject with a code value other than A, you will receive an error message. Further, if in retrieving the list of segment names to be displayed on the form, one (or more) of the related segments is found not to have a DL/I alias, the form will be displayed with an error message and the mode of processing switched to display mode.

Once entered in the table, neither the segment subject nor its role can be modified on this form. You may, however, add, erase, or modify other parameters (LEVL, PARENT, DSG) that are considered parameters of the segment-data base relationship (that is, relationship data). The PARENT parameters are maintained as a group; if you omit any subject qualifiers, the current defaults are assumed.

The MODE column is processed only for new entries in the table, and not for changed or added parameters.

To delete an entry from the list, erase the SEGMENT (user name) portion of the subject. When your entries are processed, the relationship between the specified segment and the subject data base is deleted from the Dictionary. The definitions stored in the Dictionary for the segment and the data base are not affected. Any or all of the segment entries displayed on the form at a given time may be deleted in this way.

To insert a segment within the list, use the line at the bottom that is set off from the rest of the table. Enter the segment subject and any other parameters you wish to specify in the blank line, indicate after POS BEFORE SEGMENT the segment before which the insert is to be made, and select INSERT. (If you omit any qualifiers, the current defaults are assumed.)

To move a segment already listed to another position within the list, use the same line. Enter the segment subject in the blank line new values for LEVEL and PARENT (if desired); enter after POS BEFORE SEGMENT the segment before which the segment is to be moved and select MOVE. Neither the present position of the segment nor its destination position has to be displayed in the window when the INSERT or MOVE action is performed.

If you enter an asterisk (*) in the MODE column before selecting the MOVE action, a RELOCATE WITH DEPENDENTS command will be issued for the specified segment (see "RELOCATE Command" in Chapter 3).

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
3-TOP	positions window at top of stored list
4-INSERT	inserts a segment in the stored list
5-MOVE	moves an existing segment in the stored list
6-DOWN	moves window down over stored list
9-RTN	returns to previously displayed PHYSICAL BATABASE form
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

Specialized Actions

INSERT: This action causes a subject name, typed in a special display line just above the action area, to be inserted in the stored list for this form, at the position you specify following the POS BEFORE SEGMENT field on the next line. The segment that the inserted segment is to precede need not appear on the display.

MOVE: This action moves a segment name already present in a stored list to the position specified by your entry in the POS BEFORE SEGMENT field on the next line. Neither the segment to be moved nor the segment before which it is to be moved has to appear on the display.

POS BEFORE SEGMENT: These fields in the action area do not have a number associated with them. They are always used with either the INSERT or MOVE action. The values you supply in these fields specify the name of the segment before which a segment is to be moved or inserted.

PHY-SEGMENT FORM

The PHY-SEGMENT form is used to enter and display the definition of a real (as opposed to a virtual) segment occurring in a DL/I physical data base. The form allows you to define common segment attributes, as well as the parameters of a logical parent, a real logical child, or a physical parent of a logical child. The blank form in edit mode is shown in Figure 30.

```

PHY-SEGMENT | NAME:
              | CODE:      OCC:      STAT:
DESC:
DATE:         | (MAX)BYTES:      MINBYTES:      NOFLDS:      ALIGNED:
DATABASE:     | CODE: P  STAT:   ROLE= P  FREQ:
(PHYSICAL PTRS)- PPAR:      PTWIN:      HIER:
***((FOR LOGICAL PARENT, LOGICAL CHILD, OR PHYSICAL PARENT OF LOGICAL CHILD)-
  (RULES)- INS:      DEL:      REP:      PLACE:      CTR:
***((FOR LOGICAL CHILD SEGMENT)-
  (LOGICAL PARENT)- SEGMENT:      CODE:      OCC:      STAT:
                    DATABASE:     STAT:      KEYSTRG:
  (LOGICAL PTRS)-  LPAR:      LTWIN:      PAIRED:
(E/C EXIT)- ROUTINE:      KEY/DATA:      INIT:
=====
ACT:  _  1-PROC  2-REGEN  3-DESC  4-ALIAS  5-FLDS  7-PL/I  9-REUSE 10-HDR 11-EXPL
        6-USERDATA-NO:  _  8-LCHILD SEGMENT:  _____ CODE= A OCC:  _  STAT:  _
        (RESPONSE LINE)
    
```

DISPLAY Mode:

```

=====
ACT:  _  2-REGEN  3-DESC  4-ALIAS  5-FLDS  7-PL/I  9-REUSE 10-HDR 11-EXPL
        6-USERDATA-NO:  _  8-LCHILD SEGMENT:  _____ CODE= A OCC:  _  STAT:  _
        (RESPONSE LINE)
    
```

Figure 30. PHY-SEGMENT Form

Accessing the PHY-SEGMENT Form

You can access the PHY-SEGMENT form from the HEADER form (if the segment has ROLE-P with respect to the specified data base), by an automatic transfer from the no-role SEGMENT form once the role is specified, or by selecting the REUSE action on another SEGMENT form.

Content

See "Common Fields in Segment Forms" in the "No-role SEGMENT Form" section below for an explanation of the fields at the top of the form. The remaining fields are described below:

(PHYSICAL PTRS)-

Under this field are subfields that specify the physical pointers parameters. These are the subfields.

PPAR

This field specifies whether single or double pointers are to be used in the physical parent segment for this segment type (like the values SNGL and DBLE for PARENT in the IMS/VIS DBDGEN SEGM statement). Valid characters are:

S Single
D Double

PTWIN

This field specifies whether physical twin pointers are to be used for this segment type and, if so, whether they are forward or forward and backward pointers. Valid characters are:

T Physical twin forward
B Physical twin forward and backward

HIER

This field specifies whether hierarchical pointers are to be used for this segment type and, if so, whether they are forward or forward and backward pointers. Valid characters are:

H Hierarchical forward
B Hierarchical forward and backward

(RULES)-

The following fields correspond to parameters in the RULES operand for a SEGM statement in an IMS/VIS DBDGEN specification.

INS

This field specifies the rule to be used to insert segments into the data base. Valid characters are:

P Physical
L Logical
V Virtual

DEL

This field specifies the rule to be used to delete segments from the data base. Valid characters are:

P Physical
L Logical
V Virtual
B Bi-directional

REP

This field specifies the rule to be used to replace segments in the data base. Valid characters are:

P Physical
L Logical
V Virtual

PLACE

This field specifies the rule to be used in determining where segments are to be inserted in the data base. Valid characters are:

F First

L Last

H Here

CTR

This field specifies whether a reserve counter is to be placed in the segment's prefix. A value of C specifies a reserve counter. If the value is omitted, no counter is reserved.

The (LOGICAL PARENT) and (LOGICAL PTRS) specifications are used for parameters of a segment that is a real logical child. The first four (LOGICAL PARENT) fields identify the segment that is the logical parent of the subject segment. The DATABASE and STAT fields identify the data base in which that logical parent segment occurs. (CODE=P and OCC=0 are assumed.) The last (LOGICAL PARENT) subfield is described below:

KEYSTRG

This field specifies whether a symbolic pointer to the logical parent segment is to be stored in the subject segment. Valid characters are:

P Pointer desired

V No symbolic pointer

This parameter corresponds to the second PARENT operand in a SEGM statement for IMS/VS DBDGEN.

The (LOGICAL PTRS) fields specify parameters for the pointers relating the subject segment to its logical parent:

LPAR

This field specifies whether space for a pointer to the logical parent is to be reserved in the segment prefix. A value of P reserves a logical parent pointer. If the value is omitted, the pointer is not reserved.

LTWIN

This field specifies the type of logical twin pointers for this segment. Valid characters are:

L Logical twin forward

B Logical twin forward and backward

PAIRED

This field specifies whether the segment has a bidirectional relationship. A value of P signifies the segment is paired. If the value is omitted, it is not paired.

The (E/C EXIT) fields are described in "Common Fields in Segment Forms" in the no-role SEGMENT form below.

Working with the Form

A description of the use of any fields not explained below may be found in "Common Usage of Segment Forms" in the "No-role SEGMENT Form" section below. The values specified in the (PHYSICAL PTRS) and (RULES) fields may be modified at your discretion.

In the (LOGICAL PARENT) fields, the logical parent segment subject and the data base subject are treated as groups. If qualifiers are omitted, the current defaults are assumed. Modifications to any of these fields are treated as a change in the specification of the logical parent segment or data base, and not as a change in the subjects for the logical parent segment or data base being displayed.

The values specified in the KEYSTRG and (LOGICAL PTRS) fields may be modified at your discretion.

If you modify the values specified in the NAME, OCC, and STAT fields at the top of the form or in the DATABASE row, you are expected to select the REUSE action. In this event, changes to other parts of the screen will be lost.

If you select REUSE, the form you get will depend upon whether the segment you specify has a relationship established with the data base displayed in the DATABASE field or not. If it does, the role-dependent form appropriate for that relationship will be displayed. You may then select the appropriate role-dependent form from the no-role SEGMENT form.

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
3-DESC	displays DESCRIPTION form for subject
4-ALIAS	displays ALIASES form for subject
5-FLDS	displays SEGMENT-FIELDS form
6-USERDATA_NO	selects set of User Data for subject
7-PL/I	displays PL/I DATA form for segment
8-LCHILD	displays LCHILD form
9-REUSE	displays the appropriate SEGMENT form to be used with another subject
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

Specialized Actions

LCHILD: This action displays the LCHILD (logical child) form. When you choose this action, you should enter the complete segment name in the adjacent fields, unless you want to accept the current defaults for CODE, OCC, and STAT. If the segment you entered does not have a relationship already established, the subject name will be displayed on the LCHILD form without highlighting.

PL/I: This action displays the PL/I DATA form associated with the particular segment or field defined on the form from which this action is selected.

PIP-SEGMENT FORM

The PIP-SEGMENT form is used to enter and display the definition of a primary index pointer segment (that is, the single segment occurring in a primary index data base). In addition to the common attributes you specify about a segment, the form lets you specify the single key field that must be declared within an index pointer segment. The blank form in edit mode is shown in Figure 31.

Accessing the PIP-SEGMENT Form

You can access the PIP-SEGMENT form from the HEADER form (if the subject segment has ROLE=X established), by an automatic transfer from the no-role SEGMENT form once this role has been established, or by specifying the REUSE action on a SEGMENT form.

PIP-SEGMENT	NAME:								
	CODE:	OCC:	STAT:						
DESC:									
DATE:	BYTES:					ALIGNED:			
DATABASE:	CODE: X	STAT:	ROLE= X	FREQ:					
(KEY-FIELD)-									
	NAME:	CODE= A	OCC:	STAT:					
	TYPE:	BYTES:	START:						
=====									
ACT: _	1-PROC	2-REGEN	3-DESC	4-ALIAS	7-PL/I	10-HDR	11-EXPLAIN		
	6-USERDATA-NO: _	9-REUSE							
(RESPONSE LINE)									

DISPLAY Mode:

=====									
ACT: _	2-REGEN	3-DESC	4-ALIAS	7-PL/I	10-HDR	11-EXPLAIN			
	6-USERDATA-NO: _	9-REUSE							
(RESPONSE LINE)									

Figure 31. PIP-SEGMENT Form

Content

See "Common Fields in SEGMENT Forms" in the "No-role SEGMENT Form" section below for an explanation of the fields at the top of the form. Note that only a single BYTES field is used to specify the length of the index pointer segment, since it has a fixed length.

The (KEY-FIELD) specification is used to define the single field that must be declared within the index pointer segment. It has seven labeled subfields.

The NAME, CODE, OCC, and STAT fields specify the complete Dictionary subject of the key field.

TYPE

This field specifies the type of data in the key field. Valid values are:

C	Alphameric
X	Binary
P	Packed

BYTES

This field specifies the length of the key field in bytes. The value may be 0 to 32767.

START

This field specifies the starting position of the key field in the index pointer segment. The value may be 0 to 32767. The default is 1.

Working with the Form

The BYTES field on this form corresponds to the (MAX)BYTES field on other SEGMENT forms.

In the (KEY-FIELD) specification, the subject fields are treated as a group. If the OCC or STAT qualifiers are omitted, the current subject qualifier defaults are assumed (note that CODE=A is required). If the subject entered does not already appear in the Dictionary, a new entry is created in the element category and a relationship is established (with SEQ=U) between the element definition and the definition for the subject pointer segment.

If you modify the values specified in the NAME, OCC, and STAT fields at the top of the form or in the DATABASE row, you are expected to select the REUSE action. In this event, changes to other parts of the screen will be lost.

If you select REUSE, the form you get will depend upon whether the segment you specify has a relationship established with the data base displayed in the DATABASE field or not. If it does, the role-dependent form appropriate for that relationship will be displayed. If no relationship has been established, the no-role SEGMENT form will be displayed. You may then select the appropriate role-dependent form from the no-role SEGMENT form.

If you erase (write over with minus signs) the NAME portion of the subject, the relationship between that field and the subject segment is deleted, but the definition stored in the Dictionary for the field and is not affected.

Values may only be entered in the TYPE, BYTES, and START fields if the subject has been entered. The default for START is 1. The values in these fields may be modified at your discretion.

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
3-DESC	displays DESCRIPTION form for subject
4-ALIAS	displays ALIASES form for subject
6-USERDATA_NO	selects set of User Data for subject
7-PL/I	displays PL/I DATA form for segment or field
9-REUSE	displays the appropriate SEGMENT form for use with another subject
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

Specialized Actions

PL/I: This action displays the PL/I DATA form associated with the particular segment or field defined on the form from which this action is selected.

PL/I DATA FORM

The PL/I DATA form is used to enter and display PL/I source text, comments, or narrative and acts as a window that can be moved up or down over the lines of stored PL/I data. For every subject definition in the segment or element (field) category, up to 255 lines of PL/I data may be entered. The blank display form is shown in Figure 32.

PL/I DATA		(subject Segment or Field identifier)
LNO	T	TEXT
<pre> ===== ACT: _ 1-PROC 2-REGEN 3-TOP 9-RTN 10-HDR 11-EXPLAIN 6-DOWN: 15_ (RESPONSE LINE) ===== </pre>		

DISPLAY Mode:

ACT: _		2-REGEN 3-TOP	9-RTN 10-HDR 11-EXPLAIN
6-DOWN: 15_		(RESPONSE LINE)	

Figure 32. PL/I DATA Form

Accessing the PL/I DATA Form

You can access the PL/I DATA form from any of the following forms:

FIELD	PHY-SEGMENT
LOGICAL SEGMENT	SIP-SEGMENT
no-role SEGMENT	VLC-SEGMENT
PIP-SEGMENT	

Content

The subject to which this PL/I data belongs is displayed in labeled fields at the top of the form. The subject cannot be modified on this form. The table below the subject has columns for the following entries.

LNO This field specifies the line number. The value may be from 1 to 255.

T This field specifies the type of source text. Valid characters are:

C	Comment
N	Narrative description
X	Unsupported attribute

TEXT

This field contains text of the type specified in the T field. The text may be up to 72 characters long.

Working with the Form

When the PL/I DATA form is presented, the window is positioned at the top of the PL/I data stored for this subject. You can use the DOWN action to move the window down over the text. To return to the beginning of the stored data, select the TOP action.

To enter new PL/I lines, position the window so that one or more empty lines appear at the bottom of the table. Then enter the line number, the type of source text, and the text in the appropriate columns. After you have checked your entry for accuracy, select PROC to store the new line(s) in the Dictionary. When the entries are processed, they will be moved to the positions within the stored PL/I data indicated by their line numbers (and may disappear from the display form you are viewing).

To delete a stored line of PL/I data, first position the window so that the line is displayed. Then erase the line number by crossing it out with minus signs and select PROC. To replace a line, modify the line as displayed.

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
3-TOP	positions window at top of stored list
6-DOWN	moves window down over stored list
9-RTN	returns to previously displayed form
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

PRIMARY INDEX DATABASE FORM

The PRIMARY INDEX DATABASE form is used to enter and display all parameters relevant to the definition of a DL/I primary index data base: the data base itself, the index pointer segment occurring within the data base, and the single key field that must be declared within that pointer segment. A DL/I primary index data base is a data base that is an index of root segments for a physical data base having ACCESS=HIDAM. Note that in IMS/VS the parameters specified in the (TARGET-SEG) and INDEXED-FIELD fields would appear in an LCHILD statement appended to the SEGM statement for the (INDEX-SEG) segment in the DBDGEN specification for the index data base. The blank form in edit mode is shown in Figure 33.

Accessing the PRIMARY INDEX DATABASE Form

You can access the PRIMARY INDEX DATABASE form from the HEADER form.

Content

The NAME and STAT fields at the top of the form show the subject for the data base. CODE=X and OCC=000 are assumed. For rules for these and other subject fields, see Figure 8. The form contains additional labeled fields for the following entries:

PRIMARY INDEX DATABASE		NAME:	STAT:		
DESC:					
OSACCS:	DOSCOMP:	PASSWD:			
SECURITY - QUERY:	UPDATE:	DATMUSER-NO:	DATMUSER-SEQ:		
(DATASET)- DD1:	SIZE1/BLK1:	RECL1:	DEVICE:		
OVFLW:	SIZE2/BLK2:	RECL2:	MODEL:		
(INDEX-SEG)-	NAME:	CODE= A	OCC:	STAT:	
	FREQ:	BYTES:			
(KEY-FIELD)-	NAME:	CODE= A	OCC:	STAT:	
	TYPE:	BYTES:	START:		
(TARGET-SEG)-	NAME:	CODE= A	OCC:	STAT:	
	DATABASE:	STAT:			
INDEXED-FIELD:					
=====					
ACT: _	1-PROC	2-REGEN	3-DESC	10-HDR	11-EXPLAIN
	6-USERDATA-NO: _	9-REUSE			
(RESPONSE LINE)					

DISPLAY Mode:

=====					
ACT: _	2-REGEN	3-DESC	10-HDR	11-EXPLAIN	
	6-USERDATA-NO: _	9-REUSE			
(RESPONSE LINE)					

DELETE Mode:

=====					
ACT: _	1-DELETE	9-REUSE	10-HDR	11-EXPLAIN	
(RESPONSE LINE)					

Figure 33. PRIMARY INDEX DATABASE Form

OSACCS

This field specifies the OS/VS access method used with this data base. (The DL/I access method is assumed to be INDEX.) The value may be ISAM or VSAM.

DOSCOMP

This field specifies whether DOS compatibility is to be observed. The value D specifies that DOS compatibility is desired. If omitted, DOS compatibility is not observed.

PASSWD

This field specifies whether the DBDNAME is used as a VSAM password. The value may be Y for Yes or N for No.

**SECURITY
QUERY**

This field specifies an access code to be used in reading from a data base. Any three alphameric characters are valid.

If GIS data base security is enabled, values must be numeric, from 0 to 128. The value 0 means no security access code is required.

UPDATE

This field specifies an access code to be used in updating a data base. Any three alphameric characters are valid.

If GIS data base security is enabled, values must be numeric, from 0 to 128. The value 0 means no security access code is required.

DATMUSER-NO

This field specifies which User Data set (1 to 5) is to hold specifications for the data base. There is no default value.

DATMUSER-SEQ

This field specifies the starting sequence number in the User Data set specified in the DATMUSER-NO field. (The range of available User Data lines is 1 to 999.) For GIS, the content of the User Data could be JCL statements or DATM statements.

(DATASET)

The following subfields correspond directly to DATASET statement parameters for an IMS/VS DBDGEN.

DD1

This field specifies the ddname of the primary data set of the data set group. It is required for DBD_OUT. The value may be 1 to 8 alphameric characters.

SIZE1/BLK1

This field specifies the block size for DD1. It is optional for DBD_OUT. The value may be 0 to 32767.

RECL1

This field specifies the logical record length for DD1. It is optional for DBD_OUT. The value may be 0 to 32767.

OVFLW

This field specifies the ddname of the overflow data set for the data set group. It is optional for DBD_OUT. The value may be 1 to 8 alphameric characters.

SIZE2/BLK2

This field specifies the block size for the overflow data set. The value may be 0 to 32767.

RECL2

This field specifies the logical record length for the overflow data set. It is optional for DBD_OUT. The value may be 0 to 32767.

DEVICE

This field specifies the type of physical storage device for the data set group. The typical values include 2319, 2314, 2305, 3330, 3340, 3350, 3400, 2400, or TAPE.

MODEL

This field specifies the model of the device specified in the DEVICE field. The value may be 1, 2, or 11, with the following meaning:

1	Model of 2305 or 3330
2	Model of 2305
11	Model of 3330

(INDEX-SEG)

This is a row on the form that contains NAME, CODE, OCC, and STAT fields for the single pointer segment in the index data base. The subfields FREQ and BYTES further describe the single pointer segment and are explained below.

FREQ This field specifies the number of segments of this type that are expected to occur within the data base. The value may be 1 to 8 numeric characters to the left of the decimal point plus two numeric characters to the right of the decimal. The value range is 0.01 to 16777215.

BYTES This field specifies the length of the segment. The value may be 0 to 32767.

(KEY-FIELD) This is a row on the form that contains the NAME, CODE, OCC, and STAT fields for the key field (element) in the index pointer segment. The subfields TYPE and BYTES further describe the key field and are explained below.

TYPE This field specifies the type of data within the element. Valid characters are:

C	Alphameric
X	Binary
P	Packed

BYTES This field specifies the length of the segment. The value may be 0 to 32767.

START This field specifies the starting position of the key field in the segment. The value may be 0 to 32767.

(TARGET-SEG) This is two rows on the form that contain the NAME, CODE, OCC, and STAT fields for a root segment in a HIDAM data base, and DATABASE and STAT fields for the HIDAM data base that is the target of the index. The subfield INDEXED-FIELD is described below.

INDEXED-FIELD This field specifies the name of the key field in the target root segment.

Working with the Form

This form may be used to create and edit the complete definition for a DL/I primary index data base. If the subject names you specify for the data base (on the HEADER form), the related subject (in the INDEX-SEG fields), and the related element (in the KEY-FIELD fields) are all new, the Dictionary will automatically add definitions for each subject and establish appropriate relationships between them.

If, however, one or more of the subjects is already defined in the Dictionary, complications may occur when the form is processed. These complications occur when the necessary relationships have not been established between the subjects you have specified.

The Dictionary uses relationships as access paths to data about the pointer SEGMENT and its key-field ELEMENT in the primary index data base. If these relationships don't exist, the Dictionary does not find the SEGMENT or ELEMENT subject and tries to add it together with the appropriate relationship. When it does so, it may also try to add subject data (based on your entries in the form) where subject data already exists. This attempt produces an error message.

If you receive such an error message, select the REGEN action. When the form is redisplayed, the subject name for the SEGMENT and/or ELEMENT, now linked by the relationship the Dictionary

has created, will appear in the appropriate fields in highlight. Any subject and relationship data that exists will also be displayed. At this point, you can make changes to the fields on the form and continue processing.

If you modify the values specified in the NAME and STAT fields at the top of the form, you are expected to select the REUSE action. In this event, changes to other parts of the screen will be lost.

Values displayed in the OSACCS, DOSCOMP, PASSWD, (DATASET), and all the SECURITY fields may be modified at your discretion.

Within the (INDEX-SEG) fields, the segment name fields are treated as a group; if the OCC or STAT qualifiers are omitted, the current defaults are assumed (note that CODE=A is required). Specification of the pointer segment subject results in the automatic creation of a Dictionary entry for the segment, if the segment was not defined previously, and the establishment of a relationship (with ROLE=X) between that segment and the index data base. Once entered, the subject fields cannot be modified.

To delete a relationship between a pointer segment and an index data base, erase (write over with minus signs) the NAME portion of the pointer segment subject. The relationship entry and the index pointer segment definition are both deleted, but the definition of the data base is not affected. The specification for the key field disappears from the display, but the field definition is not affected.

Values may be entered into the FREQ and BYTES fields only in the context of a complete index pointer segment subject. The values in both fields may be modified on this form.

The (KEY-FIELD) specification may also be entered if the index pointer segment subject has been entered. The subject fields must be treated as a group; if you omit the OCC or STAT qualifiers, the current defaults are assumed (note that CODE=A is required). Specification of the key field subject results in the automatic creation of a Dictionary entry for the field (if the field was not defined previously), and the establishment of a relationship (with SEQ=U) between that field definition and the definition for the pointer segment. Once entered, the subject cannot be modified.

To delete the relationship between the key field and the pointer segment, simply erase (write over with minus signs) the NAME field of the field subject and select PROC. The relationship is deleted, but the field and segment definitions are not affected. (The field definition may be deleted only on the definition form for that field.)

To enter values in the TYPE, BYTES, or START field, you must first enter the key field subject. These three fields may all be modified on this form. You may enter further specifications for the key field on the proper FIELD form. If no START value is entered, START=1 is assumed.

In the (TARGET-SEG) fields, the subject fields for the target segment and the target data base must be treated as groups; if you omit any qualifiers, the current defaults are assumed. Unlike the (INDEX-SEG) and (KEY-FIELD) display fields, the (TARGET-SEG) display fields do not allow automatic definition of a segment or data base. Any modification of these fields is treated as a change in the target of the index relationship, and not as a change in the subjects for either the segment or data base. If you erase the NAME portion of the target segment identifier, the entire target segment definition (including the INDEXED-FIELD specification) is deleted from the Dictionary.

A value may be specified in the INDEXED-FIELD field only if subject names have been entered for the index pointer segment and target segment. You may modify the value of this field on this form.

Note: On this form, you are not asked to specify if your entries are new or old definitions, so the Dictionary system cannot check your accuracy. It is assumed that you have checked your entries carefully. If you enter a slight variation of a subject currently in the Dictionary, the Dictionary assumes it is a new subject, and creates another entry.

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
3-DESC	displays DESCRIPTION form for subject
6-USERDATA_NO	selects set of User Data for subject
9-REUSE	redispays form for use with another subject
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

PROGRAM FORM

The PROGRAM form is used to enter and display Dictionary definitions for programs. The blank PROGRAM form in edit mode is shown in Figure 34.

Accessing the PROGRAM Form

You can access the PROGRAM form from the HEADER form.

Content

The NAME, OCC, and STAT fields at the top of the form show the name, occurrence number, and status code that make up the Dictionary subject name for this program. Subject code G, indicating a program, is assumed.

The form has additional labeled fields, as follows:

TYPE

This field specifies the program type. Valid values are:

B	Batch Message Processing
T	Telecommunication
D	Data base

LANG

This field specifies the programming language. Valid characters are A to Z. The following values have special meaning:

C	COBOL
P	PL/I
A	Assembler language
F	FORTRAN

SIZE

This field specifies the program size in bytes. The value may be 1 to 16777215.

PROGRAM	NAME:	OCC:	STAT:
DESC:			
TYPE:			
LANG:			
SIZE:			
=====			
ACT: _	1-PROC	2-REGEN	3-DESC
	6-USERDATA-NO: _	9-REUSE	10-HDR 11-EXPLAIN
(RESPONSE LINE)			

DISPLAY Mode:

=====			
ACT: _	2-REGEN	3-DESC	10-HDR 11-EXPLAIN
	6-USERDATA-NO: _	9-REUSE	
(RESPONSE LINE)			

DELETE Mode:

=====			
ACT: _	1-DELETE		10-HDR 11-EXPLAIN
		9-REUSE	
(RESPONSE LINE)			

Figure 34. PROGRAM Form

Working with the Form

If you modify the values specified in the NAME, OCC, and STAT fields at the top of the form, you are expected to select the REUSE action. In this event, changes to the screen below these fields will be lost.

The values in the TYPE, LANG, and SIZE fields may be modified at your discretion.

Processing Options

The options available for this form while in edit mode are:

- | | |
|---------------|---|
| 1-PROC | processes entered data |
| 2-REGEN | regenerates contents of current form |
| 3-DESC | displays DESCRIPTION form for subject |
| 6-USERDATA_NO | selects set of User Data for subject |
| 9-REUSE | redispays form for use with another subject |
| 10-HDR | returns to HEADER form |
| 11-EXPLAIN | displays an EXPLANATION form |

Enter the action number in the ACT field, or select a PF key.

PSB FORM

The PSB form is used to enter and display definitions of DL/I PSBs (Program Specification Blocks). The blank form in edit mode is shown in Figure 35.

```
PSB  NAME:          STAT:

DESC:

  LANG:          IOASIZE:          SSASIZE:
  MAXQ:          CMPAT:            OLIC:
  IOEROPN:       WTOR:

(SYSTEM DEFINITION PARAMETERS)-
  PSBTYPE:
  OVLY:
  IQF:

=====
ACT:  _  1-PROC  2-REGEN  3-DESC  4-TRAN          10-HDR  11-EXPLAIN
      6-USERDATA-NO:  _  9-REUSE
                        (RESPONSE LINE)
```

DISPLAY Mode:

```
=====
ACT:  _          2-REGEN  3-DESC  4-TRAN          10-HDR  11-EXPLAIN
      6-USERDATA-NO:  _  9-REUSE
                        (RESPONSE LINE)
```

DELETE Mode:

```
=====
ACT:  _  1-DELETE
                        9-REUSE
                        (RESPONSE LINE)
```

Figure 35. PSB Form

Accessing the PSB Form

You can access the PSB form from the HEADER form.

Content

The NAME and STAT fields at the top of the form show the PSB name and status code of the PSB. CODE=P and OCC=0 are assumed. (See Figure 8. for subject rules.)

The form has the following labeled fields:

LANG

This field specifies the programming language of programs that use this PSB. Valid characters are:

C	COBOL
P	PL/I
A	Assembler language

IOASIZE

This field specifies the largest I/O area in bytes used by the programs. The value may be 1 to 32767.

SSASIZE

This field specifies the maximum size of the SSAs used by the programs. The value may be 1 to 32767.

MAXQ

This field specifies the maximum number of data base calls with QX status codes that may be issued between synchronization points. The value may be 1 to 32767.

CMPAT

This field specifies whether an I/O PCB is always present with this PSB. Valid values are:

Y	Yes
N	No

OLIC

This field specifies whether online image copy is a valid operation when using this PSB. Valid values are Y for yes and N for no.

IOEROPN

This field specifies the condition code to be returned if an I/O error occurs. The value can be from 0 to 4095.

WTOR

This field specifies whether a WTOR is requested when an I/O error occurs. A value of W specifies that a WTOR is requested.

The (SYSTEM DEFINITION PARAMETERS) are used to specify attributes of the PSB that apply to an IMS/VS stage 1 system definition. This area contains the following labeled fields:

PSBTYPE

This field specifies whether the program is a batch or telecommunication program. Valid characters are:

B	Batch
T	Telecommunication

OVLY

This field specifies whether the program uses an overlay design. Valid values are:

Y	Yes
N	No

IQF

This field specifies whether the program is an Interactive Query Facility program. Valid values are:

Y	Yes
N	No

Working with the Form

If you modify the values specified in the NAME and STAT fields at the top of the form, you are expected to select the REUSE action. In this event, changes to the screen below these fields will be lost.

The values specified in the LANG, IOASIZE, SSASIZE, MAXQ, CMPAT, OLIC, IOEROPN, WTOR, and (SYSTEM DEFINITION PARAMETERS) fields may be modified at your discretion.

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
3-DESC	displays DESCRIPTION form for subject
4-TRAN	displays PSB-TRANSACTION form
6-USERDATA_NO	selects set of User Data for subject
9-REUSE	redispays form for use with another subject
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

Specialized Actions

TRAN: This action displays the SYSDEF-TRANSACTION form. This form displays a list of the transactions related to the PSB that is the subject of the previously selected PSB form.

PSB-TRANSACTION FORM

This form lists all the transactions that have been related to a particular PSB and allows for the specification of new relationships.

The PSB-TRANSACTION form acts as a window (with a maximum of 15 lines) that can be moved up or down over the list of transactions. Relationships may have been created with update commands, or by prior use of this, or the RELATED ENTITIES form. The blank form in edit mode is shown in Figure 36.

Accessing the PSB-TRANSACTION Form

You can access this form by selecting the TRAN action on the PSB form.

Content

The PSB and STAT fields at the top of the form contain the full subject of the PSB. Subject code P and OCC=0 are assumed. The subject cannot be modified on this form.

The table has four labeled columns. The first three, TRANSACTION, OCC, and STAT, are the used to specify the subject names of the listed transactions. The remaining column, OSTTYPE specifies whether the transaction is an input or an output transaction. The valid values are I (input) and O (output). Output transactions are ignored by the STAGE_1_OUT command.

PSB-TRANSACTION		PSB:	STAT:
TRANSACTION	OCC	STAT	OSTTYPE
=====			
ACT: _	1-PROC	2-REGEN	3-TOP
	6-DOWN: 15_		9-RTN 10-HDR 11-EXPLAIN
(RESPONSE LINE)			

DISPLAY Mode:

=====			
ACT: _	2-REGEN	3-TOP	9-RTN 10-HDR 11-EXPLAIN
	6-DOWN: 15_		
(RESPONSE LINE)			

Figure 36. PSB-TRANSACTION Form

Working with the Form

When the form is displayed, the window is positioned at the top of the list of transactions. To move the window over the list, use the DOWN action. To return to the beginning of the list, select TOP.

To enter the names of additional transactions, position the window so that one or more blank lines appear at the bottom of the form. Only the subject has to be entered (and processed) to create an entry for this relationship. If not all the subject qualifiers are entered, the current defaults are assumed.

You can only enter relationships for transactions already defined in the Dictionary. If the subject entered is not already in the transaction category, you will receive an error message.

Once entered into the table, a transaction subject cannot be modified on this form.

An entry can be deleted from the list by erasing (writing over with minus signs) the user name (TRANSACTION column) portion of the subject. On processing, the relationship between the specified transaction and subject PSB is deleted from the Dictionary, but the PSB and transaction definitions are not affected. Any or all of the transaction entries displayed on the form may be deleted in this manner.

The value specified in the OSTTYPE field is considered to be a parameter of the PSB-transaction relationship (that is, relationship data) and may be modified at your discretion.

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
3-TOP	positions window at top of stored list
6-DOWN	moves window down over stored list
9-RTN	returns to previously displayed PSB form
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

RELATED ENTITIES FORM

The RELATED ENTITIES form is used to enter and display, for a specified subject, a list of related subjects in a specified category. This form is used only for relationship entries that are not aliases and do not include relationship data. On any given access, this display form deals with relationships between one specified subject and other subjects in just one specified category, unless *SY0 is the category specified. If *SY0 is specified, all subjects in the seven system categories that are related to the subject specified on the form are listed.

The RELATED ENTITIES form is a window form that displays a stored list of related subjects. The blank form in edit mode is shown in Figure 37.

RELATED category	category:	CODE:	OCC:	STAT:
NAME	CODE	OCC	STAT	
<p>=====</p> <p>ACT: _ 1-PROC 2-REGEN 3-TOP 10-HDR 11-EXPLAIN</p> <p>6-DOWN: 15_</p> <p>(RESPONSE LINE)</p>				

DISPLAY Mode:

ACT: _	2-REGEN	3-TOP	10-HDR	11-EXPLAIN
6-DOWN: 15_	(RESPONSE LINE)			

Figure 37. RELATED ENTITIES Form

Accessing the RELATED ENTITIES Form

You can access the RELATED ENTITIES form from the HEADER form by specifying a standard category in the RELATIONSHIPS TO row. See the "Standard HEADER Form" section in Chapter 2 for more information. You may use *SYO in place of a standard category to get the related system subjects. If you are interested in only a specific system category use that category in place of *SYO.

Content

The full title of the display form depends on the category specified in the (RELATIONSHIP) line of the HEADER form. If the request was for related subjects in the segment subject category, the upper left corner reads: RELATED SEGMENTS. The subject whose relationships have been requested appears at the top center^a of the form. These labels cannot be modified by the user on this form.

The table in the center of the form has four columns for the NAME (user name), CODE (subject code), OCC (occurrence number), and STAT (status), which make up the full subject name entered or displayed.

Working with the Form

When the form is first presented, the window is positioned at the top of the list of related subjects specified by the title and subject at the top of the form. You may select the DOWN action to move the window down over the stored list.

To enter a new related subject, position the window so that one or more empty lines appear at the bottom of the table. Then enter the related subject or names. If the full subject is not entered, default qualifiers will be used. When you have checked your entry for accuracy, select PROC to cause the relationship entry to be stored in the Dictionary.

Note that some relationships shown on this form normally require relationship data (for example, segment with data base or PSB). These cases should be avoided, because the relationship data cannot be entered on the RELATED ENTITIES form and establishing the relationship without this data may not produce the desired result. A request for relationships between the specified subject and an invalid category results in an error message. When the RELATED ENTITIES form is used to display relationships that contain relationship data, the relationship data is not displayed and therefore new relationships should not be added using the RELATED ENTITIES form.

If you enter a name not already stored in the specified category of related subjects, a new definition is created for that name.

Once entered in the table (and processed), a related subject cannot be changed on this form.

An entry can be deleted from the list by erasing (that is, crossing out with minus signs) the user name portion of the subject and then selecting PROC. Upon processing, the relationship will be deleted from the Dictionary. The definitions of the two subjects involved in the relationship will not be affected. Any or all of the related subject entries displayed on a form may be deleted in this way.

Processing Options

The options available for this form while in edit mode are:

- 1-PROC processes entered data
- 2-REGEN regenerates contents of current form
- 3-TOP positions window at top of stored list
- 6-DOWN moves window down over stored list
- 10-HDR returns to HEADER form
- 11-EXPLAIN displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

RELATIONSHIP DATA FORM

This form is used to edit and display relationship data pertaining to an instance of an installation-defined relationship between two subjects. One subject must belong to a installation-defined category. The blank form in edit mode is shown in Figure 38.

RELATIONSHIP DATA	RELATIONSHIP: INVERSE NAME:	CODE: OCC: STAT: CODE: OCC: STAT:
(SUBJECT1)-category:		
(SUBJECT2)-category:		
(SEQUENCE)-xxxxxxx:	DIRECTED:	
ACT: _ 1-PROC 2-REGEN 7-UP: _ 8-DOWN: _ 9-RTN 10-HDR 11-EXPLAIN 4-SUBJECT1 5-SUBJECT2 (RESPONSE LINE)		

DISPLAY Mode:

ACT: _ 2-REGEN 7-UP: _ 8-DOWN: _ 9-RTN 10-HDR 11-EXPLAIN 4-SUBJECT1 5-SUBJECT2 (RESPONSE LINE)
--

Figure 38. RELATIONSHIP DATA Form

Accessing a RELATIONSHIP DATA Form

You can access a RELATIONSHIP DATA form from either version of the SUBJECT-RELATIONSHIPS form, having marked a particular relationship and selected the RELDATA action.

Content

The RELATIONSHIP field displays the keyword that related SUBJECT1, as the primary subject, to the subject named for SUBJECT2. The INVERSE NAME field displays the keyword used as inverse name, that relates SUBJECT2 to SUBJECT1, for reference.

Below are two lines identified by (SUBJECT1) and (SUBJECT2). These display the pair of subjects that are related. The category fields are replaced by the category names of each of the related subjects. The user name portion of the subject name follows the category name, and the CODE, OCC, and STAT fields contain the qualifier values. SUBJECT1 is regarded as the primary subject, and SUBJECT2 is the one that is related to it. The sense of direction is from SUBJECT1 to SUBJECT2.

The (SEQUENCE) field has the xxxxxxxx replaced by the name of the keyword that governs the sequencing of this type of relationship. Next to the keyword the current value for the keyword is displayed. This value may be changed but not deleted. If there is no sequence keyword, the xxxxxxxx field is replaced by (NONE) and no value is displayed.

The DIRECTED field is followed by a single character. If the principal subject, SUBJECT1, has a relationship to SUBJECT2 that has a distinct sense of dependency, then this is reflected in the value in the DIRECTED column. The meaning of the values that can occur are:

- D** The relationship is downward; SUBJECT1 is superior to SUBJECT2.
- U** The relationship is upward; SUBJECT1 is subordinate SUBJECT2.
- N** The relationship is not directed; N appears as the default value.

The body of the form is a table of keywords and any stored values that exist. Keywords with accompanying values are listed downward in the left side of the table, and continue on the top right side. A keyword's value may be up to 120 characters long. There is a fixed value field associated with each keyword. That field may be from one to five 24 character lines in the value column. A keyword and its value must always appear in the same column; so if there is not enough room at the bottom of a column for the value field, the keyword and its value will appear in the next column or on the next screen.

Working with the Form

Neither the subject names nor the relationship keywords may be modified on the form. If a value is shown for the SEQUENCE attribute, it can be changed but not deleted, because it identifies the actual relationship instance.

The relationship described by this form already exists, so the attributes of the relationship may be altered by supplying or changing values for the displayed keywords.

Since a keyword has to fit completely on the current display, the presence of one or more blank lines at the bottom of the right-hand column of keywords may not be an indication that you have reached the end of the list of keywords. It would be a good practice to look beyond the end of the list by using the DOWN action. On this form, the numbers associated with the UP and DOWN actions refer to the number of attributes, rather than to the number of lines.

If a keyword's value cannot be specified completely in the 24 positions available to each keyword line, it is continued in the next row in the Value column. Up to four continuation lines may be added. When a value does require a continuation, the value must be entered totally within the rows allotted for that

keyword. The value can **not** have embedded blanks, so that its characters extend to the column boundary, and are continued in the first available position in the succeeding line.

To delete a value, you erase the whole value by writing over it with minus signs. To change a value, write over the old data and blank out any extraneous characters.

The values for the relationship data may be checked for length or range, may be checked for inclusion in a list, or may be checked by a routine provided by your installation. Information about the keywords and the required values should have been provided by your installation in the form of additional documentation, or by annotated Dictionary Guide reports, or may be available by viewing EXPLANATION frames obtainable by selecting the EXPLAIN action.

The SUBJECT1 and SUBJECT2 actions cause the definition of the chosen subject to be displayed on the appropriate key form. You will not be able to return directly to the RELATIONSHIP DATA form if you select either of these actions.

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
4-SUBJECT1	displays appropriate key form
5-SUBJECT2	displays appropriate key form
7-UP	moves window up over list of relationships data keywords
8-DOWN	moves window down over list of relationships data keywords
9-RTN	returns to previously displayed SUBJECT-RELATIONSHIPS form
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

Specialized Actions

SUBJECT1: This action displays the subject specified in the (SUBJECT1) field on the appropriate key form.

SUBJECT2: This action displays the subject specified in the (SUBJECT2) field on the appropriate key form.

SECONDARY INDEX FORM

The SECONDARY INDEX form is used to enter and display the definition of a specific secondary index (which may share a secondary index data base with other indexes). You define the secondary index on this form by defining a secondary index pointer segment and relating it to a secondary index data base definition. The secondary index name is the name of the pointer segment. The form accepts entries for the index pointer segment, the key field within that segment, and the segment that is the "target" of the index. From this form you may also request an associated XDFLD (indexed field) form, on which you may enter specifications to complete the secondary index definition.

When you enter specifications for the index pointer segment and the key field within that segment, definitions for those subjects will be created automatically if they are not already defined.

The secondary index form and the associated XDFLD form together contain all the parameters needed to define a secondary index. These two forms and the associated SECONDARY INDEX DATABASE form include all the parameters needed to completely define a secondary index data base. The blank SECONDARY INDEX form in edit mode is shown Figure 39.

SECONDARY INDEX	DATABASE:	STAT:			
(INDEX-SEG)-	NAME:	CODE= A	OCC:	STAT:	
	FREQ:	BYTES:			WHRRULE:
(KEY-FIELD)-	NAME:	CODE= A	OCC:	STAT:	
	TYPE:	BYTES:	START:		SEQ:
(TARGET-SEG)-	NAME:	CODE= A	OCC:	STAT:	
	DATABASE:	STAT:			
	XDFLD-NAME:	PTR:			
=====					
ACT: _	1-PROC	2-REGEN	3-DELETE	8-XDFLD	9-RTN 10-HDR 11-EXPLAIN
	5-SEC INDEX NAME:		CODE= A	OCC: _	STAT: _
(RESPONSE LINE)					

DISPLAY Mode:

=====					
ACT: _	2-REGEN	8-XDFLD	9-RTN	10-HDR	11-EXPLAIN
	5-SEC INDEX NAME:		CODE= A	OCC: _	STAT: _
(RESPONSE LINE)					

Figure 39. SECONDARY INDEX Form

Accessing the SECONDARY INDEX Form

You can access the SECONDARY INDEX form from the SECONDARY INDEX DATABASE form or from a SECONDARY INDEX form for a secondary index sharing the same data base.

Content

The DATABASE and STAT fields at the top of the form show the name and status codes for a secondary index data base. These values are displayed for information only; you cannot modify them on this form.

The (INDEX-SEG) row contains NAME, CODE, OCC, and STAT fields used to define the secondary index pointer segment. The three additional subfields are described below.

FREQ

This field specifies the number of segments of this type expected to occur within the index data base. The value may be 1 to 8 numeric characters to the left of the decimal point and two numeric characters to the right of the decimal. The value range is 0.01 to 16777215.

BYTES

This field specifies the length of the segment. The value may be any decimal number from 0 to 32767.

WHRRULE

This field specifies where new occurrences of the segment are to be inserted in the data base. This field is the same as the second value for the RULES parameter. The value may be one alphabetic character:

F First

L Last

H Here

The (KEY-FIELD) row contains NAME, CODE, OCC, and STAT fields that are used to define the key field. The key field must be declared within the index pointer segment. Specification of the key field subject results in the automatic creation of a Dictionary entry for the field, and the establishment of a relationship between the key field and the index pointer segment. The four additional subfields are described below.

TYPE

This field specifies the type of data. Valid values are:

C Alphameric

X Binary

P Packed

BYTES

This field specifies the length of the field in bytes. The value is a decimal number from 0 to 32767.

START

This field specifies the starting position of the field in the segment. The value is a decimal number from 0 to 32767. The default is 1.

SEQ

This field specifies whether the field is a sequence field and, if it is, whether the key is unique or nonunique. Valid values are:

U Unique sequence field

M Nonunique sequence field

(TARGET-SEG)

The (TARGET-SEG) row contains NAME, CODE, OCC, and STAT, and DATABASE and STAT fields to identify the segment that is the target of the secondary index and the data base in which the segment occurs. CODE=P and OCC=000 are assumed for the data base subject.

XDFLD-NAME

The XDFLD-NAME field is used to specify the name of the indexed field within the segment that is the target of the secondary index. The name entered into this field appears as the INDEX= parameter in the LCHILD specification for the target segment (TARGET-SEG) associated with the relationship between the index pointer segment (INDEX-SEG) and the index data base. DL/I conventions require that the name match the name entered into the XDFLD-NAME field at the top of the associated XDFLD form.

PTR

The PTR field is used to specify the type of pointers to be placed in the index pointer segments. Valid values for this parameter are S=single or Y=symbolic.

In IMS practice, the parameters specified in the (TARGET-SEG), XDFLD-NAME, and PTR fields would appear in an LCHILD statement appended to the SEGM statement for the (INDEX-SEG) segment in the DBDGEN specification for the index data base.

Working with the Form

This form may be used to create and edit the definition for a secondary index within a shared secondary index data base; that is to say, the definitions for the index pointer SEGMENT subject and the ELEMENT subject representing its key-field. If the subject names you specify for the related SEGMENT (in the fields associated with the SEC\$INDEX\$NAME action you used when accessing the form) and the key-field ELEMENT (in the KEY-FIELD fields) are all new, the Dictionary will automatically add definitions for both subjects and establish the appropriate relationships between them and the secondary index DATABASE subject.

If, however, either the segment or the element is already defined in the Dictionary, complications may occur when the form is processed. These complications occur when the necessary relationships have not been established either between the index DATABASE and the SEGMENT you have specified, or between that SEGMENT and the ELEMENT you have specified.

The Dictionary uses relationships as access paths to data about the pointer SEGMENT and its key-field ELEMENT in the secondary index data base. If these relationships don't exist, the Dictionary does not find the SEGMENT or ELEMENT subject, and tries to add it together with the appropriate relationship. When it does so, it may also try to add subject data (based on your entries in the form) where subject data already exists. This attempt produces an error message.

If you receive such an error message, select the REGEN action. When the form is redisplayed, the subject name for the SEGMENT and/or ELEMENT, now linked by the relationship the Dictionary has created, will appear in the appropriate fields in highlight. Any subject and relationship data that exists will also be displayed. At this point, you can make changes to the fields on the form and continue processing.

The (INDEX-SEG) subject is carried over from the previous form. When you process the index segment subject, a Dictionary definition for the segment is automatically created (if the segment was not defined previously), and a relationship (with ROLE=Y for the secondary index pointer) between the segment and the index data base is established. You cannot change or erase the subject; if you wish to delete the definition for the secondary index, you must use the DELETE action.

You may enter and change the values in the FREQ and BYTES fields at your discretion.

Additional specifications for the index pointer segment may be entered on the SIP-SEGMENT form for that segment.

Within the (KEY-FIELD) specification, the subject fields must be treated as a group; if you omit the OCC or STAT qualifiers, the current defaults are assumed (note that CODE=A is required). When you enter the key field subject, a Dictionary entry for that key field is automatically created (if it does not already exist), and a relationship is established between that field definition and the definition for the pointer segment. Once entered, the subject cannot be changed.

To delete the relationship between the key field and the pointer segment, erase the NAME portion of the key field subject by crossing it out with minus signs. Note that this action does not delete the definition for the key field itself. That definition can be deleted only on the FIELD form for that field.

You can enter values in the TYPE, BYTES, START, and SEQ fields only if the key field subject has been entered. The values in these four fields may be modified at your discretion.

Additional specifications for the key field may be entered on the FIELD form for that field.

Within the (TARGET-SEG) specification, the subject fields for the target segment and the target data base are treated as groups; if you omit any qualifiers, the current defaults are assumed. (Note, however, that CODE=A is required for the target segment; CODE=P and OCC=0 are assumed for the target data base.) Any change in these fields is treated as a change in the index relationship (that is, as a naming of a different index target) and not as a change in the subject for the target segment or data base previously specified. You may enter values in the XDFLD-NAME and PTR fields only if a subject has been entered for the index target segment. The values in these fields may be modified at your discretion.

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
3-DELETE	deletes subject only; returns to SECONDARY INDEX DATABASE form (See explanation below)
5-SEC INDEX NAME	displays another SECONDARY INDEX form
8-XDFLD	displays an XDFLD form
9-RTN	returns to previously displayed form
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

Specialized Actions

DELETE: On this form the DELETE action deletes only the subject, not the definitions for the index pointer segment or the key field. After the DELETE action is processed, the SECONDARY INDEX DATABASE form from which this form was accessed is redisplayed.

SEC INDEX NAME: This action displays the SECONDARY INDEX form, which shows the definition of an index occurring in the index data base. Before you select this action, you should specify the complete subject name of the secondary index in the labeled fields provided with this action (or the user name and the current default qualifiers). If the secondary index is not yet defined in the Dictionary, it will not be highlighted when the SECONDARY INDEX form is presented.

XDFLD: This action displays the XDFLD (indexed field) form.

SECONDARY INDEX DATABASE FORM

The SECONDARY INDEX DATABASE form is used to enter and display definitions of both shared and unshared secondary index data bases, as well as parameters relevant to the physical allocation of these data bases. (The secondary index or indexes in the index data base are defined on the associated SECONDARY INDEX forms.) A DL/I secondary index data base may include from one to 16 secondary indexes for one or more target data bases. The blank form in edit mode is shown in Figure 40.

Accessing the SECONDARY INDEX DATABASE Form

You can access the SECONDARY INDEX DATABASE form from the HEADER form.

SECONDARY INDEX DATABASE	NAME:	STAT:					
DESC:							
PROT:	DOSCOMP:	PASSWD:					
SECURITY - QUERY:	UPDATE:	DATMUSER-NO:	DATMUSER-SEQ:				
(DATASET)-	DD1:	SIZE1:	RECL1:				
	OVFLW:	SIZE2:	RECL2:				
	DEVICE:	MODEL:					
=====							
ACT: _	1-PROC	2-REGEN	3-DESC	4-ALIAS	9-REUSE	10-HDR	11-EXPLAIN
	6-USERDATA-NO: _	5-SEC INDEX NAME: _____	CODE=A	OCC: _	STAT: _		
(RESPONSE LINE)							

DISPLAY Mode:

=====							
ACT: _	2-REGEN	3-DESC	4-ALIAS	9-REUSE	10-HDR	11-EXPLAIN	
	6-USERDATA-NO: _	5-SEC INDEX NAME: _____	CODE=A	OCC: _	STAT: _		
(RESPONSE LINE)							

DELETE Mode:

=====							
ACT: _	1-DELETE	9-REUSE	10-HDR	11-EXPLAIN			
(RESPONSE LINE)							

Figure 40. SECONDARY INDEX DATABASE Form

Content

The NAME and STAT fields at the top show the subject for the data base. CODE=Y and OCC=000 are assumed. In addition, the form contains labeled fields for the following entries:

PROT

This field specifies whether the PROT option is specified. It applies only to INDEX and VSAM data bases. Valid values are:

- P PROT specified
- N NO PROT specified

DOSCOMP

This field specifies whether DOS compatibility is to be observed. Specify D if DOS compatibility is desired.

If omitted, the default is that DOS compatibility is not observed.

PASSWD

This field specifies whether the DBDNAME is to be used as a VSAM password. The value is Y for yes or N for no.

**SECURITY
QUERY**

This field specifies an access code to be used in reading from a data base. Any three alphameric characters are valid.

If GIS data base security is enabled, values must be numeric, from 0 to 128. The value 0 means no security access code is required.

UPDATE

This field specifies an access code to be used in updating a data base. Any three alphameric characters are valid.

If GIS data base security is enabled, values must be numeric, from 0 to 128. The value 0 means no security access code is required.

DATMUSER-NO

This field specifies which User Data set (1 to 5) is chosen to hold data set specifications for the subject data base. There is no default value.

DATMUSER-SEQ

This field specifies the starting sequence number in the User Data set chosen by the DATMUSER-NO field value to hold data set specifications for the subject data base. (The range of available User Data lines is 1 to 999.) For GIS, the content of the User Data could be JCL statements or DATM statements.

(DATASET)

The following fields correspond to the DATASET statement parameters for an IMS/VIS DBDGEN.

DD1

This field specifies the DD name of the primary data set for this data set group. The value may be 1 to 8 alphameric characters.

SIZE1

This field specifies the block size for DD1. It is optional for DBD_OUT. The value may be 0 to 32767.

RECL1

This field specifies the logical record length for DD1. It is optional for DBD_OUT. The value may be 0 to 32767.

OVFLW

This field specifies the DD name of the overflow data set for this data set group. It is optional for DBD_OUT. The value may be 1 to 8 alphameric characters.

SIZE2

This field specifies the block size for the overflow data set. The value may be 0 to 32767.

RECL2

This field specifies the logical record length for the overflow data set. It is optional for DBD_OUT. The value may be 0 to 32767.

DEVICE

This field specifies the type of physical storage device for this data set group. It is required for DBD_OUT. Typical values include 2319, 2314, 2305, 3330, 3340, 3350, 3400, 2400, or TAPE.

MODEL

This field specifies the model of the device specified in the DEVICE field. The value may be 1, 2, or 11, with the following meaning.

1	Model of 2305 or 3330
2	Model of 2305
11	Model of 3330

Working with the Form

If you modify the values specified in the NAME or STAT fields at the top of the form, you are expected to select the REUSE action. In this event, changes to other parts of the screen will be lost.

You may also modify the values in the PROT, DOSCOMP, PASSWD, and SECURITY fields, or in any of the (DATASET) fields.

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
3-DESC	displays DESCRIPTION form for subject
4-ALIAS	displays ALIASES form for subject
5-SEC INDEX NAME	displays a SECONDARY INDEX form
6-USERDATA_NO	selects set of User Data for subject
9-REUSE	redispays form for use with another subject
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

Specialized Actions

SEC INDEX NAME: This action displays the SECONDARY INDEX form, which shows the definition of an index occurring in the index data base. Before you select this action, you should specify the complete subject name of the secondary index in the labeled fields provided with this action (or the user name and the current default qualifiers). If the secondary index is not yet defined in the Dictionary, it will not be highlighted

NO-ROLE SEGMENT FORM

General Comments on the SEGMENT Forms

In DL/I a segment is the basic unit of data that is stored and retrieved by an application program. A segment may be defined as fixed or variable in length and may contain one or more logically related data fields.

Six different key forms are used to enter and display segment definitions. These forms differ principally according to the role established for the segment in its relationship to a particular data base.

The no-role SEGMENT form is used to enter and display the definition for a segment independent of its association with any specific data base. This form is displayed when the user requests a segment definition form on the HEADER form without specifying a qualifying data base subject name.

The blank no-role SEGMENT form in edit mode is shown in Figure 41.

The no-role SEGMENT form is used mainly to develop the initial definition for a segment before its role with respect to a particular data base has been established. Note that the same segment can be related to (that is, can occur within) different data bases and have different roles in each. To establish a new relationship between a previously defined segment and a new or previously defined data base, you first request a display of the segment definition, with or without filling in the IN DATABASE field. Then, when the no-role SEGMENT form is displayed, enter the additional information. As soon as both the data base name and the segment role are entered on this form, the definition will be redisplayed on the appropriate role-dependent form.

Two of the role-dependent segment forms are designed for definitions of segments that are related to a physical data base. The other three treat the definitions for logical segments, primary index pointer segments, and secondary index pointer segments.

Common Fields in SEGMENT Forms

A common group of fields appear in nearly all of the six key segment forms. These fields, their meaning, and permissible values are discussed here. The rest of the fields on the different role-dependent forms are discussed under the descriptions of these forms.

The NAME, CODE, OCC, and STAT fields at the top of the form contain the segment subject. The user name portion of the subject for a segment may be up to 31 characters long.

The other common fields are as follows:

DATE

This field specifies any date you want to associate with the defined field. Valid values must be six digits long in the form, "mmddy". Values may range from 010100 to 123199.

(MAX)BYTES

This field specifies the maximum length for a variable-length segment or the length for a fixed-length segment. The value may be 0 to 32767.

MINBYTES

This field specifies the minimum length for a variable-length segment. The value may be 0 to 32767.

NOFLDS

This field specifies the number of fields in the segment. The value is 0 to 255. This field is not maintained by the Dictionary.

ALIGNED

This field specifies whether the segment is to be aligned. Valid values are:

Y	Yes, aligned
N	No, unaligned

The DATABASE, CODE, and STAT fields specify the Dictionary subject of the data base to which the segment is related. OCC=0 is assumed.

ROLE

This field specifies the role of the segment with respect to the specified data base. Valid values are:

P	Physical
---	----------

L	Logical
V	Virtual logical child
X	Primary index pointer
Y	Secondary index pointer

FREQ

This field specifies the number of segments of this type expected to occur for a parent. The value may have 1 to 8 digits to the left of the decimal point and two digits to the right of the decimal. The value range is 0.01 to 16777215.

(E/C EXIT)

The following fields define the edit/compression option for the segment, with ROUTINE, KEY/DATA and INIT parameters corresponding to the COMPRTN operand of the SEGM statement for IMS/V5 DBDGEN.

ROUTINE

This field specifies the name of the user-supplied edit or compression routine. The value may be 1 to 8 alphameric characters.

KEY/DATA

This field specifies whether the KEY or DATA compression option is to be used. Valid values are:

K	Key (all)
D	Data

INIT

This field specifies whether initialization and termination processing control is required. If initialization and termination processing is required, specify I, otherwise leave this field blank.

Common Usage of SEGMENT Forms

If you modify the values specified in the NAME, OCC, and STAT fields at the top of the form or in the DATABASE row, you are expected to select the REUSE action. In this event, changes to other parts of the screen will be lost.

If you select REUSE, the form you get will depend upon whether the segment you specify has a relationship established with the data base displayed in the DATABASE field or not. If it does, the role-dependent form appropriate for that relationship will be displayed. If no relationship has been established, the no-role SEGMENT form will be displayed. You may then select the appropriate role-dependent form from the no-role SEGMENT form.

Values may be entered and modified in the DATE, (MAX)BYTES, MINBYTES, and NOFLDS fields at your discretion. Note that the specification of a MINBYTES value implies that the subject segment has a variable length.

The fields of the data base subject (DATABASE, CODE, and STAT) must be treated as a group; if any of the qualifiers are omitted, the current defaults are assumed. Processing of a data base subject entered on the no-role segment form causes a relationship to be established between the definitions for the subject segment and the specified data base. If an entry for the data base does not exist, it will be added to the Dictionary. Note that the segment is placed last in the ordered list of segments related to that data base.

Once specified, the data base subject can only be modified by using the REUSE action. The DATABASE field on the SEGMENT form can, however, be erased (crossed out with minus signs); on processing, the relationship between the subject segment and

that data base is deleted from the Dictionary (without otherwise affecting the segment and data base definitions), and the segment definition is redisplayed on the SEGMENT form.

A ROLE value may only be specified within the context of a data base subject, and must be consistent with the specified data base subject code. Segments related to a physical data base (CODE=P) may have roles P or V (physical or virtual logical child). Segments related to a logical data base (CODE=L) must have ROLE=L. Segments related to a primary index data base (CODE=X) must have ROLE=X. Segments related to a secondary index data base (CODE=Y) must have ROLE=Y. Once a role has been specified on the SEGMENT form, and the information has been processed, the segment definition is immediately redisplayed on the appropriate role-dependent segment form. The specified role value cannot be modified on any of the role-dependent SEGMENT forms. On those forms in which they appear, the values specified in the FREQ fields and the fields comprising the (E/C EXIT) specification may be modified at the user's discretion.

SEGMENT	NAME:			
	CODE:	OCC:	STAT:	
DESC:				
DATE:	(MAX)BYTES:	MINBYTES:	NOFLDS:	ALIGNED:
DATABASE:	CODE:	STAT:	ROLE:	
=====				
ACT: _	1-PROC	2-REGEN	3-DESC	4-ALIAS 5-FLDS 7-PL/I 10-HDR 11-EXPLAIN
	6-USERDATA-NO: _	9-REUSE	(RESPONSE LINE)	

DISPLAY Mode:

=====				
ACT: _	2-REGEN	3-DESC	4-ALIAS	5-FLDS 7-PL/I 10-HDR 11-EXPLAIN
	6-USERDATA-NO: _	9-REUSE	(RESPONSE LINE)	

DELETE Mode:

=====				
ACT: _	1-DELETE			10-HDR 11-EXPLAIN
		9-REUSE	(RESPONSE LINE)	

Figure 41. No-role SEGMENT Form

Accessing the No-role SEGMENT Form

You can access the no-role SEGMENT form from the HEADER form or by selecting the REUSE action from another SEGMENT form.

Content

All the labeled fields displayed on the no-role SEGMENT form are explained under "Common Fields in SEGMENT Forms" in the "No-role SEGMENT Form" section above.

Working with the Form

If the form you have accessed names a data base, but no relationship has yet been established between the subject segment and that data base, or the data base is not defined in the Dictionary, the data base subject is carried over from the previous form and displayed without highlighting. If the relationship between the subject segment and the specified data base has been established, but no role has yet been specified for the segment with respect to that data base, the data base subject is displayed in highlight.

You can use this form to further define the segment by entering parameters in the labeled fields as explained under "Common Usage of SEGMENT Forms" in the "No-role SEGMENT Form" section above.

You may use this form to establish a relationship between the subject segment and a data base by entering the data base subject in the appropriate fields. If you also specify (in the ROLE field) a role for the segment with respect to the data base, and select PROC for process, the Dictionary system will redisplay the segment definition on the appropriate role-dependent SEGMENT form. If the data base specified is not already defined in the Dictionary, a definition will be added before the relationship is established.

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
3-DESC	displays DESCRIPTION form for subject
4-ALIAS	displays ALIASES form for subject
5-FLDS	displays SEGMENT-FIELDS form
6-USERDATA_NO	selects set of User Data for subject
7-PL/I	displays PL/I DATA form for segment or field
9-REUSE	displays appropriate SEGMENT form for use with another subject
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

SPECIALIZED ACTIONS

PL/I: This action displays the PL/I DATA form associated with the particular segment or field defined on the form from which the action is selected.

SEGMENT-FIELDS FORM

The SEGMENT-FIELDS form is used to enter and display the specifications for fields declared within a segment. The form acts as a window (for a maximum of 15 lines) that can be moved up and down over the stored list of fields and their specifications.

DL/I allows a maximum of 255 fields to be declared within a single segment type for a given data base. The Dictionary system, however, places no limit on the number of fields that can be related to a segment. Since no more than 255 fields for each segment can be recognized at DBD generation in the host DL/I system, only 255 fields can appear with a segment as output of the DBD_OUT command. But any number of fields per segment can appear as output of the STRUCTURES_OUT command.

Definitions of segments and fields are stored in their respective subject categories, and their relationships are entered in both categories. A relationship between a segment and a field may be established by appropriate entries on either the FIELD-SEGMENTS form or on the SEGMENT-FIELDS form. Note, however, that only the **relationship** may be entered on the FIELD-SEGMENTS form, while the SEGMENT-FIELDS form also allows automatic definition of a field as its relationship to a segment is entered.

The blank form in edit mode is shown in Figure 42.

Accessing the SEGMENT-FIELDS Form

You can access the SEGMENT-FIELDS form by selecting the FLDS action from any SEGMENT form except the PIP-SEGMENT form.

Content

The SEGMENT, CODE, OCC, and STAT fields at the top of the form contain the subject for the segment this form refers to. The subject of the segment cannot be modified on this form. The table has columns for these entries:

The FIELD, CODE, OCC, and STAT fields specify the complete Dictionary subject for the field.

TYPE

This field specifies the type of data in the field. Valid values are:

B	Binary digits
C	Characters
D	Long floating point
E	Short floating point
F	Fixed point binary (word)
H	Fixed point binary (halfword)
P	Packed decimal
X	Hexadecimal digits
Z	Zoned decimal

LEN

This field specifies the field length in bytes. The value may be from 0 to 32767.

SEGMENT-FIELDS	SEGMENT:	OCC:	STAT:							
FIELD	CODE	OCC	STAT	TYPE	LEN	START	BST	SEQ/GEN	MODE	
<pre> ===== ACT: _ 1-PROC 2-REGEN 3-TOP 9-RTN 10-HDR 11-EXPLAIN 6-DOWN: 15_ (RESPONSE LINE) ===== </pre>										

DISPLAY Mode:

<pre> ===== ACT: _ 2-REGEN 3-TOP 9-RTN 10-HDR 11-EXPLAIN 6-DOWN: 15_ (RESPONSE LINE) ===== </pre>									
--	--	--	--	--	--	--	--	--	--

Figure 42. SEGMENT-FIELDS Form

START

This field specifies the starting position of the field within the segment. The value may be from 1 to 32767. If the value is omitted, the default is 1.

BST

This field specifies the starting bit position in the START byte. Valid values are 1 to 8. The bit start specification is invalid for COBOL subjects.

SEQ/GEN

This field specifies whether the key field contains a unique value, or it specifies whether the field appears with the segment in DBD_OUT. Valid values are:

- null No
- U Unique values
- M Multiple values
- G Yes

Note that a sequence field (value U or M) always appears with the segment in DBD_OUT.

MODE

This field specifies whether this is a previously defined or a new field. Valid values are:

- N New (The default is N.)

0 Old

Note that the first six columns are concerned with specifications for the field itself. START and SEQ/GEN are relationship data concerned with the functions of this field within this segment. The use of MODE in checking the accuracy of your entries is explained below.

Working with the Form

When this form is first presented, the window is positioned at the top of the stored list of fields. You may use the DOWN action to move the window down over the list. To return to the beginning of the list, choose TOP.

To add a relationship with a field and at the same time create a definition for the field in the Dictionary, first position the window so that one or more blank lines appear at the bottom of the table. Then enter in a blank line the subject (and any additional specifications), enter N for new in the MODE column, and select PROC for processing. If you omit any qualifiers, the current defaults are assumed. If there is already a definition for a field with the name you entered stored in the Dictionary, you will receive an error message.

You may also use this form to add relationships between the subject segment and fields already defined in the Dictionary. First position the window so that one or more blank lines appear at the bottom of the table. Then key in the subject of the field, enter O for "old" in the MODE column, and select PROC. If you omit any qualifiers, the current defaults are assumed. If no definition exists in the Dictionary for a field with this subject, you will receive an error message. If the field exists, after processing, the form will be presented again with all the stored values displayed in highlight.

You should not enter anything into the TYPE or BYTES column when entering a field name and a mode of "O" (old). After this entry has been processed, the stored values for TYPE and/or BYTES will appear on the screen and may be modified. If data is entered for TYPE and/or BYTES when adding an "old" field to the list, message DBD0021 may appear, indicating that TYPE or BYTES had already been added. In this case, the data on the screen should be "blanked out" prior to selecting the next action, or the same message will reappear.

To delete a stored relationship, first position the window so that the related field is displayed. Then erase the user name of the field by writing over it with minus signs, and select PROC for process. On processing, the relationship between the specified field and the subject segment is deleted from the Dictionary (and from the displayed table). The stored definitions for the field and segment in question are not otherwise affected. Any or all of the field relationships displayed on the form at a given time may be deleted in this way.

You may also use this form to add specifications to fields already listed in the table (for example, to fields just declared on the SEGMENT form from which you requested this form). Simply key in the specifications in the correct row and column and select PROC for processing. You may also modify specifications already appearing in the START and SEQ/GEN columns. Note that modifications in the START field may cause the entry for the field to move within the list of fields related to the subject segment (after processing it may disappear from the current display). You cannot modify values in the subject column.

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
3-TOP	positions window at top of stored list
6-DOWN	moves window down over stored list
9-RTN	returns to previously displayed form
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

SIP-SEGMENT FORM

The SIP-SEGMENT form is used to enter and display the definition of a secondary index pointer segment (that is, a segment occurring within a secondary index data base). In addition to the usual Dictionary specifications for a segment, the form permits you to define the key field that must be declared within an index pointer segment. The blank form in edit mode is shown in Figure 43.

Accessing the SIP-SEGMENT Form

You can access the SIP-SEGMENT form from the HEADER form if the subject segment has ROLE=Y (that is, if the specified data base is a secondary index data base), by automatic transfer from the no-role SEGMENT form, or by selecting the REUSE action on another SEGMENT form.

Content

The usual segment specification fields are treated as described under "Common Fields in SEGMENT Forms" in the no-role SEGMENT form above. Note that, since the segment has a fixed length, a single BYTES field is used to specify the length. The (KEY-FIELD) specifications are used to define the single key field that must be declared within the index pointer segment.

Working with the Form

The common segment definition fields on this form are treated as described under "Common Usage of SEGMENT Forms" in the "No-role SEGMENT Form" section above. The BYTES field corresponds to the (MAX)BYTES field.

Within the (KEY-FIELD) specifications, the subject fields are treated as a group; if the OCC or STAT qualifiers are omitted, the current defaults are assumed (note that CODE=A is required). Entry of a new key-field subject automatically creates a Dictionary entry for the field and pointer segment definition. The subject of the field cannot subsequently be modified. If you subsequently erase the user name (NAME field) by writing over it with minus signs, and select PROC, the relationship between that field and the subject segment is deleted, but the field and segment definitions are otherwise unaffected.

Values can be entered in the TYPE, BYTES, START and SEQ fields only if the (KEY-FIELD) subject is complete. The values in these fields may be modified at your discretion.

If you modify the values specified in the NAME, OCC, and STAT fields at the top of the form or in the DATABASE row, you are expected to select the REUSE action. In this event, changes to other parts of the screen will be lost.

SIP-SEGMENT	NAME:			
	CODE:	OCC:	STAT:	
DESC:				
DATE:	BYTES:		NOFLDS:	ALIGNED:
DATABASE:	CODE: Y	STAT:	ROLE= Y	FREQ:
(KEY-FIELD)-				
	NAME:	CODE= A	OCC:	STAT:
	TYPE:	BYTES:	START:	SEQ:
(E/C EXIT)-				
	ROUTINE:	KEY/DATA:	INIT:	
=====				
ACT: _	1-PROC	2-REGEN	3-DESC	4-ALIAS
	5-FLDS	7-PL/I	10-HDR	11-EXPLAIN
	6-USERDATA-NO: _	9-REUSE	8-XDFLD	
	(RESPONSE LINE)			

DISPLAY Mode:

=====				
ACT: _	2-REGEN	3-DESC	4-ALIAS	5-FLDS
	7-PLI	10-HDR	11-EXPLAIN	
	6-USERDATA-NO: _	9-REUSE	8-XDFLD	
	(RESPONSE LINE)			

Figure 43. SIP-SEGMENT Form

If you select REUSE, the form you get will depend upon whether the segment you specify has a relationship established with the data base displayed in the DATABASE field or not. If it does, the role-dependent form appropriate for that relationship will be displayed. If no relationship has been established, the no-role SEGMENT form will be displayed. You may then select the appropriate role-dependent form from the no-role SEGMENT form.

Processing Options

The options available for this form while in edit mode are:

- | | |
|---------------|--|
| 1-PROC | processes entered data |
| 2-REGEN | regenerates contents of current form |
| 3-DESC | displays DESCRIPTION form for subject |
| 4-ALIAS | displays ALIASES form for subject |
| 5-FLDS | displays SEGMENT-FIELDS form |
| 6-USERDATA_NO | selects set of User Data for subject |
| 7-PL/I | displays PL/I DATA form for the segment |
| 8-XDFLD | displays XDFLD form |
| 9-REUSE | displays appropriate SEGMENT form for use with another subject |
| 10-HDR | returns to HEADER form |
| 11-EXPLAIN | displays an EXPLANATION form |

Enter the action number in the ACT field or select a PF key.

Specialized Actions

XDFLD: This action displays the XDFLD (indexed field) form.

PL/I: This action displays the PL/I DATA form associated with the particular segment or field defined on the form from which the action is selected.

SELECTED SUBJECT-RELATIONSHIPS FORM

This form is used to edit and display a list of the installation-defined types of relationships in which the specified (principal) subject participates. Only relationships with subjects in the specified TO-CATEGORY are listed. The form is a window form and may display up to 14 relationships in one frame.

The form is required as a means of maintaining relationships between a subject in an installation-defined category and subjects in standard categories. The blank form in edit mode is shown in Figure 44.

SUBJECT-RELATIONSHIPS		category:							
		CODE:		OCC:		STAT:			
TO-CATEGORY:									
RELATIONSHIP	SUBJECT2-NAME			CODE	OCC	STAT	SEQUENCE	MARK	
<p>=====</p> <p>ACT: _ 1-PROC 2-REGEN 3-TOP 6-DOWN: 14_ 9-RTN 10-HDR 11-EXPLAIN</p> <p>7-RELDATA 8-SUBJECT2 5-CAT-RELATIONSHIPS</p> <p>(RESPONSE LINE)</p>									

DISPLAY Mode:

<p>=====</p> <p>ACT: _ 2-REGEN 3-TOP 6-DOWN: 14_ 9-RTN 10-HDR 11-EXPLAIN</p> <p>7-RELDATA 8-SUBJECT2 5-CAT-RELATIONSHIPS</p> <p>(RESPONSE LINE)</p>									
---	--	--	--	--	--	--	--	--	--

Figure 44. Selected SUBJECT-RELATIONSHIPS Form

Accessing the Selected SUBJECT-RELATIONSHIPS Form

You can access the selected SUBJECT-RELATIONSHIPS form from the HEADER form, or from the EXTENSIBILITY SUBJECT form.

Content

The top of the form gives the subject name of the principal subject with its associated OCC, STAT, and CODE values. The category is replaced by that subject's category. The code value will be blank if the subject belongs to an installation-defined category. The TO-CATEGORY field contains the category in which all the related subjects must occur.

The body of the form is a table that lists the relationship keywords and subjects in the TO-CATEGORY that are related to primary subject. In those cases where the relationship is sequenced, the sequencing value is placed in the SEQUENCE column to the right. Columns for specifying OCC, STAT, and CODE values are included on the form. Only specify a value for CODE when the TO_CATEGORY is not installation-defined.

The rightmost column, headed MARK, is used when entering data on a row. A value of N, the default, indicates that the related subject is newly defined to the Dictionary, and that subsequent processing requires adding that subject to the Dictionary as well as the relationship to it. A value of 0 denotes that the specified subject already exists.

The MARK column is also used in conjunction with the RELDATA and SUBJECT2 action. See "Working with the Form" below for details.

Working with the Form

The table displays a list of all of the relationships between the principal subject and subjects that occur in any other category you specify. The list includes any relationships to subjects in standard categories, if the principal subject was an instance of a installation-defined category. To add relationships between subjects in installation-defined categories and subjects in standard categories, you must use this form.

To add a new relationship for the subject at the top of the form, enter the appropriate information on a blank line, including the N or 0 in the MARK column. If the new entry is for a relationship that is sequenced, a value must be given for the sequence attribute in the SEQUENCE column. To delete a relationship, enter minus signs over the relationship keyword.

You may only change the value in the SEQUENCE column. Write over the old value with a new one. When the form is processed, the item whose SEQUENCE value you change may change position in the list.

The MARK column is also used with the RELDATA and SUBJECT2 actions. By placing an * in the MARK column next to an item on the list and selecting the RELDATA or SUBJECT2 action, you can access the RELATIONSHIP DATA form (RELDATA action) or the appropriate key form (SUBJECT2 action) for the item marked. The subject indicated by the * in the MARK column is the subject of the key form. Only the topmost * is processed.

The use of the CAT-RELATIONSHIPS action enables you to obtain additional information about the different types of relationships that the principal subject may participate in. The action displays the CATEGORY-RELATIONSHIPS form, which contains a list of potential relationships for the principal subject. Such information would help you complete new items in the list on your return to the current form.

If the form was accessed from the EXTENSIBILITY SUBJECT form, then the RTN action returns to that form, still showing the current subject. If the form was accessed from the HEADER form, RTN returns to that form.

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
3-TOP	positions window at top of stored list
5-CAT-RELATIONSHIPS	displays CATEGORY-RELATIONSHIPS form
6-DOWN	moves window down over stored list
7-RELDATA	displays RELATIONSHIP-DATA form
8-SUBJECT2	displays appropriate key form
9-RTN	returns to previously displayed EXTENSIBILITY SUBJECT or HEADER form
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field or select a PF key.

Specialized Actions

CAT-RELATIONSHIPS: This action accesses the CATEGORY-RELATIONSHIPS form. This form contains a list of the types of relationships the principal subject may participate in.

RELDATA: This action displays the RELATIONSHIP DATA for the relationship marked by the * in the MARK column.

SUBJECT2: This action displays the definition for the related subject marked by the * in the MARK column on the appropriate key form.

UNSELECTED SUBJECT-RELATIONSHIPS FORM

This form is used to edit and display a list of the relationships that exist between the specified (primary) subject and subjects in all installation-defined categories. The form acts as a window, with up to 14 such relationships displayed at one time. The blank form in edit mode is shown in Figure 45.

Accessing the Unselected SUBJECT-RELATIONSHIPS Form

You can access the unselected SUBJECT-RELATIONSHIPS form from the HEADER form or from the EXTENSIBILITY SUBJECT form.

Content

The top of the form gives the subject name of the principal subject along with its associated OCC, STAT, and CODE values. The category is replaced by that subject's category. The code value will be blank if the subject belongs to an installation-defined category.

The body of the form is a table. The first column contains the relationship keyword. The second column contains any subjects in any installation-defined categories related to the primary subject. In those cases where the relationship is sequenced, the sequencing value is placed in the column to the right.

The rightmost column, headed MK, is used to specify the mode when entering data on a row. A value of N, the default, indicates that the related subject is newly defined to the Dictionary, and that subsequent processing requires adding that subject to the Dictionary as well as the relationship to it. A

SUBJECT-RELATIONSHIPS		category:				
		CODE:	OCC:	STAT:		
RELATIONSHIP	CATEGORY	SUBJECT2-NAME	OCC	STAT	SEQUENCE	MK
<pre> ===== ACT: _ 1-PROC 2-REGEN 3-TOP 6-DOWN: 14_ 9-RTN 10-HDR 11-EXPLAIN 7-RELDATA 8-SUBJECT2 5-CAT-RELATIONSHIPS (RESPONSE LINE) ===== </pre>						

DISPLAY Mode:

<pre> ===== ACT: _ 2-REGEN 3-TOP 6-DOWN: 14_ 9-RTN 10-HDR 11-EXPLAIN 7-RELDATA 8-SUBJECT2 5-CAT-RELATIONSHIPS (RESPONSE LINE) ===== </pre>						
---	--	--	--	--	--	--

Figure 45. Unselected SUBJECT-RELATIONSHIPS Form

value of 0 denotes that the related subject already exists.

The MK column is also used in conjunction with the RELDATA and SUBJECT2 action. See "Working with the Form" below for details.

Working with the Form

The table lists all of the relationships between the principal subject and subjects that occur in installation-defined categories. The list will not include any relationships to subjects in standard categories. Relationships between subjects that are both in standard categories are displayed on the RELATED ENTITIES form. Relationships between a primary subject in an installation-defined category and subjects in a standard category are displayed on the selected SUBJECT-RELATIONSHIPS form.

This form is a window form. You may use the DOWN action to move the window down over the list of relationships.

To add a new relationship for the subject at the top of the form, enter the appropriate information on a blank line, including the N or O in the MK column. If the new entry is a relationship that is sequenced, a value must be given for the sequence attribute in the SEQUENCE column.

To delete a relationship, enter minus signs over the relationship keyword.

The only field that can be changed for the displayed data is the value in the SEQUENCE column. After processing, the relationship entry may change position in the list.

The MK (or mark) column is also used with the RELDATA or SUBJECT2 actions. By placing an * in the MK column next to an item on the list and selecting the RELDATA or SUBJECT2 action you can access the RELATIONSHIP DATA form (RELDATA action) or the EXTENSIBILITY SUBJECT form (SUBJECT2 action) for the item marked. The subject indicated by the * in the MK column is the subject of the EXTENSIBILITY SUBJECT form. Only the topmost * is processed. If you select the SUBJECT2 action, you will not be able to return directly to this particular SUBJECT-RELATIONSHIPS form.

The use of the CAT-RELATIONSHIPS action enables you to obtain additional information about the different types of relationships that the principal subject may participate in. The action displays the CATEGORY-RELATIONSHIPS form, which contains a list of potential relationships for the principal subject. Such information would help you complete new items in the list and obtain additional information about the relationships that the subject of the form could have with instances of subjects in other installation-defined categories.

If the form was accessed from the EXTENSIBILITY SUBJECT form, then the RTN action will return to that form, still showing the same subject. If the form was accessed from the HEADER form, RTN returns to that form.

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
3-TOP	positions window at top of stored list
5-CAT-RELATIONSHIPS	displays CATEGORY-RELATIONSHIPS form
6-DOWN	moves window down over stored list
7-RELDATA	displays RELATIONSHIP-DATA form
8-SUBJECT2	displays EXTENSIBILITY SUBJECT form
9-RTN	returns to previously displayed EXTENSIBILITY SUBJECT or HEADER form
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

Specialized Actions

CAT-RELATIONSHIPS: This action accesses the CATEGORY-RELATIONSHIPS form. This form contains a list of the types of relationships the principal subject may participate in.

RELDATA: This action displays the RELATIONSHIP DATA form for the relationship marked by the * in the MK column.

SUBJECT2: This action displays the EXTENSIBILITY SUBJECT form for the subject marked by the * in the MK column.

SUBORDINATE/SUPERIOR FIELDS FORM

The SUBORDINATE/SUPERIOR FIELDS form is used to enter and display a list of other fields to which a given field is related. The list can show fields that are subordinate to the given named field, or fields that are superior to it. In a Dictionary entry for a field-field relationship, one field is declared to be the superior field (for example, the given field contains the other), or the field is subordinate (for example, it is contained within another field). In any given access to the form, either the superior or the subordinate related fields are displayed. The SUBORDINATE/SUPERIOR FIELDS form acts as a window (with a maximum of 15 lines) on a list of related fields. The blank form in edit mode is shown in Figure 46.

Accessing the SUBORDINATE/SUPERIOR FIELDS Form

You can access the SUBORDINATE/SUPERIOR FIELDS form from a FIELD form by entering SUP for superior or SUB for subordinate in the RELATED FIELDS field and then selecting the RELATED FIELDS action.

XXXXXXXXXXXX-FIELDS	SUBJECT-FIELD:						
	CODE:	OCC:	STAT:				
RELATED-FIELD	CODE	OCC	STAT	RCODE	START	BST	MODE
<p>=====</p> <p>ACT: _ 1-PROC 2-REGEN 3-TOP 9-RTN 10-HDR 11-EXPLAIN</p> <p>6-DOWN: 15_</p> <p>(RESPONSE LINE)</p>							

DISPLAY Mode:

=====							
ACT: _	2-REGEN	3-TOP	9-RTN 10-HDR 11-EXPLAIN				
6-DOWN: 15_	(RESPONSE LINE)						

Figure 46. SUBORDINATE/SUPERIOR FIELDS Form

Content

When the form is displayed, the title in the upper left is preceded by the word SUPERIOR or SUBORDINATE, according to the request that resulted in the display. The SUBJECT-FIELD, CODE, OCC, and STAT fields at the top contain the full subject for the field that you requested. The display form title and the subject cannot be modified on this form. The table has columns for the following entries:

The RELATED-FIELD, CODE, OCC, and STAT fields specify the complete Dictionary subject of the related field.

RCODE

This field specifies the relationship between this field and the subject field. Any alphabetic character is valid and may be assigned a meaning by your installation. The following characters have a special meaning to the Dictionary.

- C Contains
- D Depends

R	COBOL REDEFINES
6	COBOL level 66 RENAMES (START=1) or RENAMES_THRU (START=32767)
8	COBOL level 88 condition names

START

This field specifies the starting position of one field within the other; the value is given in bytes. Valid values for RCODE=C are 1 to 32767.

BST

This field specifies the starting bit position in the START byte. The valid values are 1 to 8. The bit start specification is invalid for COBOL subjects.

MODE

This field specifies whether the subject field is new or existing. Valid values are N for new and 0 for old.

Working with the Form

When the form is first presented, the window is positioned at the top of the list of fields related to the subject field. You may use the DOWN action to move the window down over the list. To return to the beginning of the list, select TOP.

To create a new entry in the table of related fields, position the window so that one or more blank lines appear at the bottom of the table. Then key in the subject and RCODE. (If you omit any subject qualifiers, the established defaults are assumed.) If this is a new field not yet defined in the Dictionary, enter N in the MODE column (or accept the default), and select PROC for process. The Dictionary checks to see if there is an existing field definition with the subject name you entered. If so, a diagnostic message will appear at the bottom of the display form. If this is found to be a new subject, the appropriate field definition and relationship entries are made in the element category.

If the related field whose name you are entering is already defined in the Dictionary, enter 0 in the MODE column and select PROC for process. The Dictionary checks to be sure there is an existing definition for the field whose subject name you entered. If none is found, you will receive an error message. If the subject is found, the relationship you entered is stored in the Dictionary.

On successful processing of your entry or entries, the display form is redisplayed with stored entries now in highlight. Note, however, that since related subordinate fields are stored in the order of their start position, new entries may disappear from the window currently displayed.

Once entered in the table, a subject cannot be modified. The values specified in the RCODE and START columns are relationship data. They specify the relationship between the fields and are not part of the separate field definitions. These values may be modified at your discretion.

To delete an entry from the list, erase the RELATED-FIELD (that is the user name) portion of the subject by writing over it with minus signs. On processing, the relationship between the specified field and the subject field is deleted from the Dictionary, and the entire entry removed from the displayed list. (The definition for the previously related field remains in the Dictionary.) Any or all of the related field entries displayed on the form at a given time may be deleted in this way.

Processing Options

The options available for this form while in edit mode are:

- 1-PROC processes entered data
- 2-REGEN regenerates contents of current form
- 3-TOP positions window at top of stored list
- 6-DOWN moves window down over stored list
- 9-RTN returns to previously displayed FIELD form
- 10-HDR returns to HEADER form
- 11-EXPLAIN displays an EXPLANATION form

Enter the action number in the ACT field or select a PF key.

SYSDEF FORM

The SYSDEF form is used to enter and display definitions for the SYSDEF category. The blank form in edit mode is shown in Figure 47.

SYSDEF	NAME:	
	OCC:	STAT:
DESC:		
=====		
ACT: _	1-PROC	2-REGEN 3-DESC 4-PSB 5-TRAN 7-DB 10-HDR 11-EXPLAIN
	6-USERDATA-NO: _	9-REUSE
	(RESPONSE LINE)	

DISPLAY Mode:

=====		
ACT: _	2-REGEN	3-DESC 4-PSB 5-TRAN 7-DB 10-HDR 11-EXPLAIN
	6-USERDATA-NO: _	9-REUSE
	(RESPONSE LINE)	

DELETE Mode:

=====		
ACT: _	1-DELETE	9-REUSE 10-HDR 11-EXPLAIN
	(RESPONSE LINE)	

Figure 47. SYSDEF Form

Accessing the SYSDEF Form

You can access the SYSDEF form from the HEADER form.

Content

The NAME, OCC, and STAT fields at the top are the name, occurrence number, and status code that make up the subject of this system definition. Subject code N is assumed for all system-definition definitions.

Working with the Form

If you modify the values specified in the NAME, OCC, and STAT fields at the top of the form, you are expected to select the REUSE action. In this event, changes to other parts of the screen will be lost.

(Note that a system-definition subject cannot have an alias.)

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
3-DESC	displays DESCRIPTION form for subject
4-PSB	displays SYSDEF-PSB form
5-TRAN	displays SYSDEF-TRANSACTION form
6-USERDATA_NO	selects set of User Data for subject
7-DB	displays SYSDEF-DATABASE form
9-REUSE	redispays form for use with another subject
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

Specialized Actions

DB: This action displays the SYSDEF-DATABASE form, which displays a list of data bases. These data bases are related to the system definition that is the subject of the SYSDEF definition form from which this action is selected.

PSB: This action displays the SYSDEF-PSB form, which displays a list of the PSBs. These PSBs are related to the system definition that is the subject of the SYSDEF definition form from which this action is selected.

TRAN: This action displays the SYSDEF-TRANSACTION form. This form displays a list of the transactions related to the system definition that is the subject of the previously selected SYSDEF form.

SYSDEF-DATABASE FORM

The SYSDEF-DATABASE form lists all the data bases that have been related to a particular system definition and allows for specification of new relationships.

The SYSDEF-DATABASE form acts as a window (with a maximum of 15 lines) that can be moved up or down over the list of data bases. The blank form in edit mode is shown in Figure 48.

Accessing the SYSDEF-DATABASE Form

You can access the SYSDEF-DATABASE form by selecting the DB action on the SYSDEF form.

SYSDEF-DATABASE				SYSTEM-DEFINITION:	
				OCC:	STAT:
DATABASE	CODE	STAT	DMBRES		
===== ACT: _ 1-PROC 2-REGEN 3-TOP 9-RTN 10-HDR 11-EXPLAIN 6-DOWN: 15_ (RESPONSE LINE)					

DISPLAY Mode:

===== ACT: _ 2-REGEN 3-TOP 9-RTN 10-HDR 11-EXPLAIN 6-DOWN: 15_ (RESPONSE LINE)					
---	--	--	--	--	--

Figure 48. SYSDEF-DATABASE Form

Content

The SYSTEM-DEFINITION, OCC, and STAT fields at the top of the form contain the full subject for this system definition. The subject cannot be modified with this form. The table has four labeled columns. The first three, DATABASE, CODE, and STAT are for the subject of the listed data bases. The remaining column is for the following entry:

DMBRES

Specifies whether the data base DMB is permanently resident. Valid values are Y for yes and N for no.

Working with the Form

When the form is first displayed, the window is positioned at the top of the data base list. Use the DOWN action to move the window down over the list. To return to the beginning of the list, select TOP.

To enter the names of additional data bases, position the window so that one or more blank lines appear at the bottom of the form. Only the subject has to be entered (and processed) to create an entry for this relationship. If not all the subject qualifiers are entered, the current defaults are assumed.

On this form, you can only enter relationships for data bases already defined in the Dictionary. If the subject entered is not already in the data base category, you will receive an error message.

Once entered into the table, a data base subject cannot be modified on this form.

An entry may be deleted from the list by erasing (writing over with minus signs) the user name (DATABASE column) portion of the subject name. In processing, the relationship between the specified data base and the subject system definition is deleted from the Dictionary, but the system-definition and data-base definitions are not otherwise affected. Any or all of the data base entries displayed on the form at a given time may be deleted in this manner.

The value specified in the DMBRES field is considered to be a parameter of the system-definition-data base relationship (that is, relationship data) and may be modified at your discretion.

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
3-TOP	positions window at top of stored list
6-DOWN	moves window down over stored list
9-RTN	returns to previously displayed SYSDEF form
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

SYSDEF-PSB FORM

The SYSDEF-PSB form lists all the PSBs that have been related to a particular system definition and allows for specification of new relationships.

The SYSDEF-PSB form acts as a window (with a maximum of 15 lines) that may be moved up or down over the list of PSBs related to the subject system definition. The blank form in edit mode is shown in Figure 49.

Accessing the SYSDEF-PSB Form

You can access the SYSDEF-PSB form by selecting the PSB action on the SYSDEF form.

Content

The SYSTEM-DEFINITION, OCC, and STAT fields at the top of the form contain the full subject for this system definition. Subject code N is assumed for all system definitions. The subject cannot be modified on this form. The table has seven labeled columns. The first two, PSB and STAT, are for the subject of the listed PSBs (CODE=P and OCC=0 are assumed). The remaining five columns are for entries as follows:

PSBLOAD

This field specifies whether the PSB is permanently resident or dynamically loaded.

R	Resident (RES)
D	Dynamically loaded (DOPT)

SYSDEF-PSB		SYSTEM-DEFINITION:					STAT:	
PSB	STAT	PSBLOAD	CLASS	SCHDTYP	RSYSID	LSYSID		
<pre> ===== ACT: _ 1-PROC 2-REGEN 3-TOP 9-RTN 10-HDR 11-EXPLAIN 6-DOWN: 15_ (RESPONSE LINE) ===== </pre>								

DISPLAY Mode:

<pre> ===== ACT: _ 2-REGEN 3-TOP 9-RTN 10-HDR 11-EXPLAIN 6-DOWN: 15_ (RESPONSE LINE) ===== </pre>								
---	--	--	--	--	--	--	--	--

Figure 49. SYSDEF-PSB Form

CLASS

This field specifies a default class from 1 to 255 for the related transactions.

SCHDTYP

This field specifies whether or not the program can be scheduled in multiple regions concurrently.

P Multiple regions (PARALLEL)

S Single region (SERIAL)

RSYSID

This field specifies the remote system identification number for this PSB in this system definition.

LSYSID

This field specifies the local system identification number for this PSB in this system definition.

Working with the Form

When the form is first displayed, the window is positioned at the top of the list of PSBs. Use the DOWN action to move the window down over the list. To return to the beginning of the list, select TOP.

To enter the names of additional PSBs, first position the window so that one or more blank lines appear at the bottom of the form. Only the subject has to be entered (and processed) to create an entry for this relationship. If STAT is not entered, the current default is assumed.

On this form, you can only enter relationships for PSBs already defined in the Dictionary. If the subject entered is not already in the PSB subject category, you will receive an error message.

Once entered into the table, a PSB subject cannot be modified on this form.

An entry may be deleted from the list by erasing (writing over with minus signs) the user name (PSB column) portion of the subject name. On processing, the relationship between the specified PSB and the subject system definition is deleted from the Dictionary, but the PSB and system-definition definitions are not otherwise affected. Any or all of the PSB entries displayed on the form at a given time may be deleted in this manner.

The values specified in the PSBLOAD, CLASS, SCHDTYP, RSYSID, and LSYSID fields are considered to be parameters of the system-definition-PSB relationship (that is, relationship data) and may be modified at your discretion.

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
3-TOP	positions window at top of stored list
6-DOWN	moves window down over stored list
9-RTN	returns to previously displayed SYSDEF form
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

SYSDEF-TRANSACTION FORM

The SYSDEF-TRANSACTION form lists all the transactions that have been related to a particular system definition and allows for specification of new relationships. (Note that each transaction must also be related to a PSB for the STAGE_1_OUT command to execute properly.)

This relationship form acts as a window (with a maximum of 14 lines) that may be moved up or down over the list of transactions. The blank form in edit mode is shown in Figure 50.

Accessing the SYSDEF-TRANSACTION Form

You can access the SYSDEF-TRANSACTION form by selecting TRAN on the SYSDEF form.

Content

The SYSTEM-DEFINITION, OCC, and STAT fields at the top of the form contain the full subject for this system definition. Subject code N is assumed. The subject cannot be modified on this form. The table has twelve labeled columns. The first three, TRANSACTION, OCC, and STAT, are for the subject name of the listed transactions. (See Figure 8. for subject rules.) The remaining nine columns are for entries as follows:

PRTYNORM

This field specifies the priority of the transaction when the number of input transactions waiting is less than the limit count. Valid values are 0 to 14.

SYSDEF-TRANSACTION			SYSTEM-DEFINITION:								
			OCC:				STAT:				
TRANSACTION	OCC	STAT	PRTY NORM	PRTY LIMT	LIMIT- CONT	MSG- CLASS	PROC- NUM	PROC- SECS	PARLIM	SCHD	SPA- KEEP
<pre> ===== ACT: _ 1-PROC 2-REGEN 3-TOP 9-RTN 10-HDR 11-EXPLAIN 6-DOWN: 14_ (RESPONSE LINE) ===== </pre>											

DISPLAY Mode:

<pre> ===== ACT: _ 2-REGEN 3-TOP 9-RTN 10-HDR 11-EXPLAIN 6-DOWN: 14_ (RESPONSE LINE) ===== </pre>											
--	--	--	--	--	--	--	--	--	--	--	--

Figure 50. SYSDEF-TRANSACTION Form

PRTYLIMT

This field specifies the priority of the transaction when the number of input transactions waiting is equal to or greater than the limit count. Valid values are 0 to 14.

LIMTCONT

This field specifies the limit count for PRTYNORM and PRTYLIMT parameters. Valid values are 1 to 65535.

MSGCLASS

This field specifies the class for this transaction code. Valid values are 1 to 255.

PROCNUM

This field specifies the maximum number of messages that can be processed for each scheduling of the transaction. Valid values are 1 to 65535.

PROCSECS

This field specifies the maximum number of seconds allowed to process each message. Valid values are 1 to 65535.

PARLIM

This field specifies the threshold value used to determine when to schedule another region for this transaction. Valid values are 1 to 32767.

SCHD

This field specifies the scheduling option for other transactions when this transaction is not schedulable. The value may be 1, 2, 3, or 4.

SPAKEEP

This field specifies where the SPA should be kept. Valid values are:

C	Core
D	DASD

Working with the Form

When the form is first displayed, the window is positioned at the top of the list of transactions. Use the DOWN action to move the window down over the list. To return to the beginning of the list, select TOP.

To enter the names of additional transactions to which this system definition is related, position the window so one or more blank lines appear at the bottom of the form. Only the subject has to be entered (and processed) to create an entry for this relationship. If not all the subject qualifiers are entered, the current defaults are assumed.

On this form you can only enter relationships for transactions already defined in the Dictionary. If the subject entered is not already in the transaction category, you will receive an error message.

Once entered into the table, a transaction subject cannot be modified on this form.

An entry may be deleted from the list by erasing (writing over with minus signs) the user name (TRANSACTION column) portion of the subject name. On processing, the relationship between the specified transaction and the subject system definition is deleted from the Dictionary, but the transaction and system-definition definitions are not otherwise affected. Any or all of the transaction entries displayed on the form at a given time may be deleted in this manner.

The values specified in the PRYTNORM, PRYTLIMIT, LIMITCONT, MSGCLASS, PROCNUM, PROCSECS, PARMLIM, SCHD, and SPAKEEP fields are considered to be parameters of the transaction-system-definition relationship (that is, relationship data) and may be modified at your discretion.

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
3-TOP	positions window at top of stored list
6-DOWN	moves window down over stored list
9-RTN	returns to previously displayed SYSDEF form
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

SYSTEM FORM

The SYSTEM form is used to enter and display system definitions. The blank form in edit mode is shown in Figure 51.

Accessing the SYSTEM Form

You can access the SYSTEM form from the HEADER form.

```
=====
SYSTEM | NAME:
        | OCC:   STAT:
DESC:
=====
ACT:  _  1-PROC  2-REGEN  3-DESC                      10-HDR  11-EXPLAIN
        6-USERDATA-NO:  _  9-REUSE
                               (RESPONSE LINE)
=====
```

DISPLAY Mode:

```
=====
ACT:  _           2-REGEN  3-DESC                      10-HDR  11-EXPLAIN
        6-USERDATA-NO:  _  9-REUSE
                               (RESPONSE LINE)
=====
```

DELETE Mode:

```
=====
ACT:  _  1-DELETE                      9-REUSE          10-HDR  11-EXPLAIN
                               (RESPONSE LINE)
=====
```

Figure 51. SYSTEM Form

Content

The NAME, OCC, and STAT fields at the top of the form reflect the user name, occurrence number, and status code for this subject. The subject code is assumed to be S=system for this category. The user name for the system may be 31 characters long.

Working with the Form

If you modify the values specified in the NAME, OCC, and STAT fields at the top of the form, you are expected to select the REUSE action. In this event, changes to the screen below these fields will be lost.

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
3-DESC	displays DESCRIPTION form for subject
6-USERDATA_NO	selects set of User Data for subject
9-REUSE	redispays form for use with another subject
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

TRANSACTION FORM

The TRANSACTION form is used to enter and display transaction definitions. Dictionary definitions in the transaction category include anticipated attributes with special relevance to IMS/VS, for example, logical terminal type. But a given installation may decide to include other types of transactions in this category. The blank form in edit mode is shown in Figure 52.

Accessing the TRANSACTION Form

You can access the TRANSACTION form from the HEADER form.

Content

The NAME, OCC, and STAT fields at the top of the form show the name, occurrence number, and status code of the subject for this transaction. Subject code T=transaction is assumed.

TRANTYPE

This field is used to specify the transaction type. Permissible values are L (LTERM or logical terminal) or T (transaction code).

LTERMTYPE

This field is used to specify that the logical terminal type (TRANTYPE=L) is a master terminal. Enter M if it is a master terminal; if not, leave it blank.

The (SYSTEM-DEFINITION PARAMETERS) are used to specify attributes of the transaction that apply to an IMS/VS stage I system definition. This area contains labeled fields for the following entries:

WFI

This field specifies whether the transaction can remain resident and wait for input. Valid values are Y for Yes and N for No.

MODE

This field specifies whether data base buffers are to be flushed with each input request or wait until program termination. Valid values are:

S	Each input request
M	Program termination

INQTRAN

This field specifies whether this is an inquiry transaction. Valid values are Y for Yes and N for No.

INQRECV

This field specifies whether an inquiry transaction is to be recovered during an emergency or normal restart. Valid values are Y for Yes and N for No.

TRANSACTION	NAME:	OCC:	STAT:
DESC:			
TRANTYPE:			
LTERMTYPE:			
(SYSTEM DEFINITION PARAMETERS)-			
WFI:	MODE:	INQTRAN:	INQRECV:
SEGNO:	SEGSIZE:	SPASIZE:	SPAFIX:
MSGSEG:	MSGRESP:	EDITTRAN:	EDITNAME:
=====			
ACT: _	1-PROC	2-REGEN	3-DESC
	6-USERDATA-NO: _	9-REUSE	10-HDR 11-EXPLAIN
(RESPONSE LINE)			

DISPLAY Mode:

=====			
ACT: _	2-REGEN	3-DESC	10-HDR 11-EXPLAIN
	6-USERDATA-NO: _	9-REUSE	
(RESPONSE LINE)			

DELETE Mode:

=====			
ACT: _	1-DELETE		10-HDR 11-EXPLAIN
		9-REUSE	
(RESPONSE LINE)			

Figure 52. TRANSACTION Form

SEGNO

This field specifies maximum output segments per IMS/VS Get Unique call. The value may be from 0 to 65535.

SEGSIZE

This field specifies the maximum number of bytes for an output segment. The value may be from 0 to 65535.

SPASIZE

This field specifies the size of the SPA. The range is 16 to the tracksize of the device used to store the SPA.

SPAFIX

This field specifies whether control will remain in transactions with the same SPA size during conversation processing. Valid values are Y for Yes and N for No.

MSGSEG

This field specifies whether the input message is one or more than one segment. Valid values are:

M Multiple

S Single

MSGRESP

This field specifies whether input from the terminal should be inhibited until a response is sent back. Valid values are Y for Yes and N for No.

EDITTRAN

This field specifies whether the transaction should be translated to all uppercase or to upper- and lowercase. Valid values are:

U Uppercase

L Upper- and lowercase

EDITNAME

This field specifies the name of the transaction input edit routine. The name may be from 1 to 8 characters long.

Working with the Form

If you modify the values specified in the NAME, OCC, and STAT fields at the top of the form, you are expected to select the REUSE action. In this event, changes to the screen below these fields will be lost.

The values specified in the TRANTYPE, LTERMTYPE, and (SYSTEM-DEFINITION PARAMETERS) fields may be modified at your discretion.

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of current form
3-DESC	displays DESCRIPTION form for subject
6-USERDATA_NO	selects set of User Data for subject
9-REUSE	redispays form for use with another subject
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

USER DATA FORM

The USER DATA form is used to enter information into and display information from one of the five User Data sets. Each set of User Data may consist of up to 999 lines containing up to 80 characters each. The Dictionary system allows freeform text entries in these user data sets. That is, the information entered is not restricted in form or content, although each line must have a unique line number from 001 to 999.

Your installation may choose between two versions of the USER DATA form. The first version of the form acts as a window for a maximum of 15 lines of User Data. Seventy-two characters of data are allowed per line. Positions 73 to 80 are reserved for a sequence number that is generated by the Dictionary. If additional lines are entered, new serial numbers are assigned. (Any existing text in positions 73 to 80 will be written over.)

The second version is a window form for a maximum of 7 lines. Each line of 80 characters is preceded by its line number (LNO). In this second version of the form, all 80 characters of data are displayed, and no serializing is performed for the user.

The two versions of the USER DATA form, 15 and 7 lines, are shown in Figure 53 and Figure 54 respectively.

USER DATA X		CATEGORY:		
		CODE:	OCC:	STAT:
LNO	TEXT			
=====				
ACT: _	1-PROC	2-REGEN	3-TOP	9-RTN 10-HDR 11-EXPLAIN
	6-DOWN: 15_			
(RESPONSE LINE)				

DISPLAY Mode:

=====				
ACT: _	2-REGEN	3-TOP	9-RTN 10-HDR 11-EXPLAIN	
	6-DOWN: 15_			
(RESPONSE LINE)				

Figure 53. USER DATA Form

Accessing the USER DATA Form

You can access the USER DATA form by filling in the number (1 to 5) of the User Data set desired and selecting the USERDATA-NO action on a previously selected key form.

Content

The set of User Data to which this form refers is identified by the appropriate number (1 to 5) after the display form title. The subject to which this User Data refers is displayed in appropriately labeled fields at the top of the form. The table has two columns, headed LNO (line number) and TEXT. User data already stored in the Dictionary is displayed in these columns in line-number order. This form cannot be used to modify the User Data number or the subject.

Working with the Form

When the form is first presented, the window is positioned at the TOP of the requested set of User Data. You may use the DOWN action to move the window down the form. If you do not specify a down count, the number will default to 15, or, for the alternative USER DATA form, to 7. To return to the first line, select TOP.

USER DATA X		CATEGORY:	CODE:	OCC:	STAT:
LNO	TEXT				
<pre> ===== ACT: _ 1-PROC 2-REGEN 3-TOP 9-RTN 10-HDR 11-EXPLAIN 6-DOWN: 7_ (RESPONSE LINE) ===== </pre>					

DISPLAY Mode:

<pre> ===== ACT: _ 2-REGEN 3-TOP 9-RTN 10-HDR 11-EXPLAIN 6-DOWN: 7_ (RESPONSE LINE) ===== </pre>					
--	--	--	--	--	--

Figure 54. Alternative USER DATA Form

To enter an additional line or lines of User Data, position the window to obtain the number of blank lines you need at the bottom of the table. Then enter the line number(s) in the LNO column, followed by the text in the TEXT column.

You do not have to assign consecutive numbers to successive lines and you may want to skip numbers to leave space for future insertions. When you have checked your entry, and corrected it if necessary, select the PROC action to store the new User Data. The newly stored lines will be moved within the stored User Data to the positions indicated by their line numbers (and may disappear from the display you are viewing).

To delete a line of User Data, position the window so that the line is displayed. Then erase the line number by crossing it out with minus signs, and select PROC. To replace a line, modify the line as displayed, and select PROC.

You may make any number of additions, deletions, or changes before you select PROC. All will be processed by the same PROC action.

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of form
3-TOP	positions window at top of stored list
6-DOWN	moves window down over stored list
9-RTN	returns to previously displayed key form
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

VLC-SEGMENT FORM

The VLC-SEGMENT form is used to enter and display the definition for a physical segment (that is, a segment in a physical data base) with the role of a virtual logical child. Although this kind of segment is not really a physical segment, it is treated as a physical segment within the context of the physical data base definition.

The blank form in edit mode is shown in Figure 55.

VLC-SEGMENT	NAME:			
	CODE:	OCC:	STAT:	
DESC:				
DATE:	(MAX)BYTES:	MINBYTES:	NOFLDS:	ALIGNED:
DATABASE:	CODE: P	STAT:	ROLE= V	
(SOURCE)-				
SEGMENT:	CODE= A	OCC:	STAT:	
DATABASE:	STAT:			
=====				
ACT: _	1-PROC	2-REGEN	3-DESC	4-ALIAS
	5-FLDS	6-USERDATA-NO: _	7-PL/I	8-REUSE
			9-REUSE	10-HDR
				11-EXPLAIN
(RESPONSE LINE)				

DISPLAY Mode:

=====				
ACT: _	2-REGEN	3-DESC	4-ALIAS	5-FLDS
	6-USERDATA-NO: _	7-PL/I	8-REUSE	9-REUSE
				10-HDR
				11-EXPLAIN
(RESPONSE LINE)				

Figure 55. VLC-SEGMENT Form

Accessing the VLC-SEGMENT form

You can access the VLC-SEGMENT form from the HEADER form (if the segment has ROLE=V established with respect to the specified data base), by an automatic transfer from the no-role SEGMENT form when the V role is established, or by selecting the REUSE action on another SEGMENT form.

Content

The fields in the upper portion of the form are for the most part as described under the "Common Fields in SEGMENT Forms" section in the "No-role SEGMENT Form" above. Note that the FREQUENCY parameter is not applicable to a virtual logical child segment. The (SOURCE) fields identify the real physical segment (SEGMENT, CODE, OCC, and STAT) to which the virtual subject segment corresponds, and the data base (DATABASE and STAT; CODE=P and OCC=0 are assumed) in which that segment occurs.

Working with the Form

The common fields are treated as described in "Common Usage in SEGMENT Forms" in the "No-role SEGMENT Form" section above.

The (SOURCE) SEGMENT fields and DATABASE fields must be treated as groups. If OCC or STAT qualifiers are omitted, the current defaults are assumed (note, however, that CODE=A is required for the source segment; CODE=P and OCC=0 are assumed for the source data base).

If you modify the values specified in the NAME, OCC, and STAT fields at the top of the form or in the DATABASE row, you are expected to select the REUSE action. In this event, changes to other parts of the screen will be lost.

If you select REUSE, the form you get will depend upon whether the segment you specify has a relationship established with the data base displayed in the DATABASE field or not. If it does, the role-dependent form appropriate for that relationship will be displayed. If no relationship has been established, the no-role SEGMENT form will be displayed. You may then select the appropriate role-dependent form from the no-role SEGMENT form.

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of form
3-DESC	displays DESCRIPTION form for subject
4-ALIAS	displays ALIASES form for subject
5-FLDS	displays SEGMENT-FIELDS form
6-USERDATA_NO	selects set of User Data for subject
7-PL/I	displays PL/I DATA form for segment
9-REUSE	displays appropriate SEGMENT form for use with another subject
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Specialized Actions

PL/I: This action displays the PL/I DATA form associated with the particular segment or field defined on the form from which the action is selected.

XDFLD FORM

The XDFLD form is used to enter and display information about a secondary index that corresponds to that within an XDFLD statement for DBDGEN. In the Dictionary, these parameters are stored with the definition for the secondary index pointer segment. They are not considered part of the target segment specification, as they are in DL/I.

The blank form in edit mode is shown in Figure 56.

XDFLD	XDFLD-NAME:				
(INDEX-SEG)-	NAME:	CODE:	OCC:	STAT:	
	DATABASE:	STAT:			
(TARGET-SEG)-	NAME:	CODE= A	OCC:	STAT:	
	DATABASE:	STAT:			
(SOURCE-SEG)-	NAME:				
	CONST:	NULLVAL:	EXTRTN:		
	FIELD-1	FIELD-2	FIELD-3	FIELD-4	FIELD-5
SRCH					
SUBSEQ					
DDATA					
=====					
ACT: _	1-PROC	2-REGEN		9-RTN	10-HDR 11-EXPLAIN
(RESPONSE LINE)					

DISPLAY Mode:

=====					
ACT: _		2-REGEN		9-RTN	10-HDR 11-EXPLAIN
(RESPONSE LINE)					

Figure 56. XDFLD Form

Accessing the XDFLD form

You can access the XDFLD form from the associated SECONDARY INDEX form to complete the secondary index definition, or from the SIP-SEGMENT form (since the indexed field information is relevant to the format of the fields within the pointer segment).

Content

The XDFLD-NAME field shows the indexed field name assigned for the secondary index. The name entered into the XDFLD-NAME field corresponds to the NAME parameter on an XDFLD statement. The (INDEX-SEG) and (TARGET-SEG) fields are for the subjects of the secondary index pointer segment and the segment that is the target of the secondary index, plus their associated data base subjects. The (SOURCE-SEG) field is for the name of the source segment for the secondary index, which is assumed to be in the same data base as the target segment and may actually be the same segment. The form also includes three labeled fields

corresponding to the CONST, NULLVAL, and EXTRTN parameters in the XDFLD statement for DBDGEN and a table for lists of field names that would be specified with the SRCH, SUBSEQ, and DDATA parameters in the XDFLD statement for DBDGEN. Their contents are as follows:

CONST

This field specifies the constant that precedes each index entry key (for shared indexes). Valid values are:

C'a' Any valid character except blank

X'nn' Any valid hexadecimal value except X'40'

NULLVAL

This field specifies the conditions under which the index entry is suppressed.

Valid values are:

C'a' Any valid character except C'B' or C'Z'

X'nn' Any valid hexadecimal value except X'C2' or X'E9'

B Blank

Z Zero

EXTRTN

This field specifies the name of the routine that processes changes in the index data base resulting from changes in the indexed data base.

The value may be 1 to 8 alphameric characters. This field is optional for DBD_OUT.

SRCH

This field specifies the user name of a field regarded as the source data for an index data base. The value may be 1 to 8 alphameric characters.

SUBSEQ

This field specifies the user name of the SUBSEQ field in the index source segment. The value may be 1 to 8 alphameric characters.

DDATA

This field specifies the user name of the DDATA field in the index source segment. The value may be 1 to 8 alphameric characters.

Working with the Form

When you access this form from the SECONDARY INDEX form, values are carried over from that form and displayed in the (INDEX-SEG) and (TARGET-SEG) fields, and their associated DATABASE and STAT fields.

When you have accessed this form from the SIP-SEGMENT form for the index pointer segment, you may enter a new XDFLD specification. Values are carried over from the SIP segment form and displayed in the (INDEX-SEG) fields, and the remaining fields on the form will be empty.

The (INDEX-SEG) subject cannot be modified on this form with either mode of access.

The values in the XDFLD-NAME, CONST, NULLVAL, and EXTRTN fields may be modified at your discretion.

The (TARGET-SEG) subject fields and the associated DATABASE and STAT fields must be treated as subject groups; if you omit any qualifiers, the current defaults are assumed (CODE=P and OCC=0 are assumed for the data base). Any change you make in these

fields is treated as a redirection of the index relationship, and not as a change in the subjects for either the segment or data base.

The value in the (SOURCE-SEG) NAME field may be modified or erased at your discretion, as can any of the field names in the table at the bottom of the form.

Processing Options

The options available for this form while in edit mode are:

1-PROC	processes entered data
2-REGEN	regenerates contents of form
9-RTN	returns to previously displayed form
10-HDR	returns to HEADER form
11-EXPLAIN	displays an EXPLANATION form

Enter the action number in the ACT field, or select a PF key.

CHAPTER 4. DICTIONARY BATCH EXECUTION

This chapter describes the operating instructions for the two host operating systems in batch mode. Operating instructions for online usage may be found in Chapter 2 of the DB/DC Data Dictionary Administration and Customization Guide. You should note that the data set names used in the JCL statements given here are default library names created by Dictionary installation procedures. Device addresses in the JCL statements are examples only; substitute addresses appropriate for your installation. Note also that DBDIMSBV is used in the JCL examples. It is assumed that the DBDIMSBV link edit is available.

BATCH OPERATION UNDER OS/VS

An example of JCL statements used to operate the Dictionary in the batch mode under IMS/VS (DB) appears in Figure 57.

```
//XXXX      JOB    user parameters
//STEP1     EXEC  PGM=DFSRR00,
//           PARM=(DLI,DBDIMSBV,DBDIMSO,,00),
//           REGION=xxxxk
//STEPLIB   DD    DSN=IMSVS.RESLIB,DISP=SHR
//           DD    DSN=IMSVS.PGMLIB,DISP=SHR
//DFSRESLB  DD    DSN=IMSVS.RESLIB,DISP=SHR
//IMS       DD    DSN=IMSVS.PSBLIB,DISP=SHR
//           DD    DSN=IMSVS.DBDLIB,DISP=SHR
//IEFRDER   DD    user parameters
//DDPSBLIB  DD    DSN=IMSVS.PSBLIB,DISP=SHR
//DDDBDLIB  DD    DSN=IMSVS.DBDLIB,DISP=SHR
//PLIPDS    DD    user parameters
//PLISEQ    DD    user parameters
//CBIPDS    DD    user parameters
//CBISEQ    DD    user parameters
//SORTWK01  DD    UNIT=SYSDA,SPACE=(CYL,(5),,CONTIG)
//SORTWK02  DD    UNIT=SYSDA,SPACE=(CYL,(5),,CONTIG)
//SORTWK03  DD    UNIT=SYSDA,SPACE=(CYL,(5),,CONTIG)
//SYSOUT    DD    SYSOUT=A
//DDLST     DD    SYSOUT=A
//DDPUNCH   DD    SYSOUT=B
//DTEDD     DD    DSN=DBD.DTEDBS,DISP=OLD
//SEGDD     DD    DSN=DBD.SEGDBS,DISP=OLD
//DBSDD     DD    DSN=DBD.DBSDBS,DISP=OLD
//PCBDD     DD    DSN=DBD.PCBDBS,DISP=OLD
//SYSDD     DD    DSN=DBD.SYSDBS,DISP=OLD
//EXTDD     DD    DSN=DBD.EXTDBS,DISP=OLD
//DDINPUT   DD    *
           Dictionary input goes here.
/*
```

Figure 57. OS/VS JCL for Batch Dictionary Operation

Note that the region size depends on your system. Alternatively, you may use the DLIBATCH procedure, described in the IMS/VS System Programming Reference Manual, by replacing //STEP1 in Figure 57 with

```
//STEP1 EXEC  DLIBATCH,MBR=DBDIMSBV,
//           PSB=DBDIMSO,REGION=xxxxK
```

and making necessary changes in the job stream, such as omitting STEPLIB, IEFRDER, and IMS DD cards.

Operating Requirements for PLI_IN Command

The PLI_IN command may be used to add PL/I data to the Dictionary. Detailed information about the command is found in Chapter 6.

In the job stream shown in Figure 57no'. two DD statements are included for PLI_IN: //PLIPDS for PL/I source from a partitioned data set (PDS), and //PLISEQ for PL/I source from a sequential data set. In a specific run, you may need one or both of these statements. If the run does not include PLI_IN commands, neither of these statements is necessary.

If the //DDINPUT data stream contains a PLI_IN command that specifies the keyword MEMBER= (or MEM= or BOOK=), the data set referred to in the //PLIPDS DD statement will be searched for the indicated PDS member.

If the //DDINPUT data stream contains a PLI_IN command that does not specify the keyword MEMBER= (or MEM= or BOOK=), the data set referred to in the //PLISEQ DD statement is opened and read as a sequential data set.

Both PLIPDS and PLISEQ ddnames may be specified in the same run.

A single PLI_IN command causes an entire sequential file to be processed. If the command stream for a single execution contains multiple PLI_IN commands referring to the sequential data set defined by the //PLISEQ DD statement, the data set will be read and processed as many times as specified. This repeated addition of a segment or element to the data base will result in its occurrence number being incremented by one to maintain the uniqueness of subjects.

Note: If DD DUMMY is coded on the //PLISEQ statement, a BLKSIZE parameter value equal to a multiple of 80 should be specified.

Operating Requirements for the COBOL_IN Command

In Figure 57 two DD statements are included for COBOL_IN: //CBIPDS for COBOL source from a partitioned data set (PDS), and //CBISEQ for COBOL source from a sequential data set. In a specific run you may need one or both of these statements. If the run does not include COBOL_IN commands neither of these statements is necessary.

If the //DDINPUT data stream contains a COBOL_IN command that specifies the keyword MEMBER= (or MEM= or BOOK=), the data set referred to in the //CBIPDS DD statement will be searched for the indicated PDS member.

If the //DDINPUT data stream contains a COBOL_IN command that does not specify the keyword MEMBER= (or MEM= or BOOK=), the data set referred in the //CBISEQ DD statement is opened and read as a sequential data set.

Both CBIPDS and CBISEQ may be specified in the same run.

A single COBOL_IN command causes the entire sequential file to be processed. If the command stream for a single execution contains multiple COBOL_IN commands referring to the sequential data set defined by the //CBISEQ DD statement, the data set will be read and processed as many times as specified. This repeated addition of a segment or element to the data base will result in its occurrence number (subject name qualifier) being incremented by one to maintain the uniqueness of subject names.

Note: If DD DUMMY is coded on the CBISEQ statement, a BLKSIZE parameter value equal to a multiple of 80 should be specified.

Operating Requirements for Sorted Glossary Output

With the REPORT command you can obtain a sorted list of subjects that match search criteria. This option requires that SORT DD statements be present in the batch execution JCL.

Four DD cards are included for glossary reports. //SORTWK0n statements with n = 1,2,3 are included to define work areas when the SORT option is used. //SYSOUT is used by the SORT routine to return messages concerning its processing. If a run does not include glossary report commands or SORT=NO has been specified option, these cards can be removed.

Although three sort work areas are defined in the job stream, you can specify as many as are needed by your installation. Refer to OS/VS Sort/Merge Programmer's Guide for guidelines in determining the work area space needed.

Your installation might also require a //SORTLIB DD statement to specify the Sort program library.

BATCH OPERATION WITH DL/I DOS/VIS

The Dictionary operates in batch mode with DL/I DOS/VIS. Figure 58 shows the JCL necessary to start a batch run with DL/I DOS/VIS.

```
// JOB          SAMPLE
// LBLTYP       NSD(01)
// DLBL        SORTWK1,,0,DA
// EXTENT      SYS010,DD3LIB,1,0,3620,20
// ASSGN       SYS010,3330,VOL=DD3LIB,SHR
// ASSGN       SYS007,3330,VOL=DD3LIB,SHR
// DLBL        DTEDD,'DBD.DTEDBS',,VSAM
// EXTENT      SYS007,DD3LIB
// DLBL        SEGDD,'DBD.SEGDBS',,VSAM
// EXTENT      SYS007,DD3LIB
// DLBL        DBSDD,'DBD.DBSDBS',,VSAM
// EXTENT      SYS007,DD3LIB
// DLBL        PCBDD,'DBD.PCBDBS',,VSAM
// EXTENT      SYS007,DD3LIB
// DLBL        SYSDD,'DBD.SYSDBS',,VSAM
// EXTENT      SYS007,DD3LIB
// DLBL        EXTDD,'DBD.EXTDBS',,VSAM
// EXTENT      SYS007,DD3LIB
// DLBL        IJSYSCL,'DBD.COREI'
// EXTENT      SYSCLB
// ASSGN       SYSCLB,3330,VOL=DD3LIB,SHR
// UPSI        00000010
// EXEC        DLZRRC00,SIZE=xxxxK
DLI,DBDIMSBV,DBDIMSO,1,HDBFR=(10)
Dictionary input goes here.
/*
```

Figure 58. DOS/VIS JCL for Batch Dictionary Execution

If the Dictionary's checkpoint capability is to be used and tape is to be used for the DL/I system log, the following statements must be added;

```
// ASSGN       SYS011,X'CUU'
// TLBL        LOGOUT
```

and bit 6 of the UPSI statement must be equal to 0.

For more information, see the DL/I DOS/VIS Operator's Reference Manual and Messages and Codes.

Operating Requirements for the COBOL_IN Command

If the input stream includes a COBOL_IN command that specifies the BOOK= (or MEMBER= or MEM=) keyword, a private or system source statement library must be assigned to SYSSLB, as shown below:

```
// ASSGN    SYSSLB,X'233'  
// DLBL     IJSYSSL,'PRIVATE COBOL SL'  
// EXTENT   SYSSLB
```

If the input stream includes a COBOL_IN command that does not specify the BOOK= (or MEMBER= or MEM=) keyword, the sequential file on disk or tape containing the COBOL source code to be processed must be assigned to SYS004, with a filename of CBISEQ, as shown below:

```
// ASSGN    SYS004,X'233'  
// DLBL     CBISEQ,'PAYROLL COBOL SOURCE'  
// EXTENT   SYS004
```

If the COBOL source code follows the COBOL_IN command in the input stream, only the following assignment is required:

```
// ASSGN    SYS004,SYSIPT
```

In this case, the end of the COBOL source may be delimited by a card with ENDCBI* or CBIEND* in columns 1 to 7 or a card with /* in columns 1 and 2.

Operating Requirements for PLI_IN Command

If the input stream includes a PLI_IN command that specifies the BOOK= (or MEMBER= or MEM=) keyword, a private or system source statement library must be assigned to SYSSLB, as shown below:

```
// ASSGN    SYSSLB,X'233'  
// DLBL     IJSYSSL,'PRIVATE PL/I SL'  
// EXTENT   SYSSLB
```

If the input stream includes a PLI_IN command that does not specify the BOOK= (or MEMBER= or MEM=) keyword, the sequential file on disk or tape containing the PL/I source to be processed must be assigned to SYS005 with a filename of PLISEQ, as shown below:

```
// ASSGN    SYS005,X'233'  
// DLBL     PLISEQ,'PAYROLL PL/I SOURCE'  
// EXTENT   SYS005
```

If the PL/I source follows the PLI_IN command in the input stream, only the following assignment is required:

```
// ASSGN    SYS005,SYSIPT
```

In this case, the end of the PL/I statements may be delimited by a card with ENDPLI* or PLIEND* in columns 1 to 7 or a card with /* in columns 1 and 2.

Operating Requirements for Sorted Glossary Output

With the REPORT command you can obtain a sorted list of subjects that match search criteria. This option requires that SORT statements be present in the batch execution JCL.

Four statements are included for glossary reports. If a run does not include glossary report commands or SORT=NO has been specified, these cards can be removed.

```
// LBLTYP NSD(01)
// DLBL SORTWK1,,0,DA
// EXTENT SYS010,DD3LIB,1,0,3620,20
// ASSGN SYS010,3330,VOL=DD3LIB,SHR
```

The Sort program is assumed to be in the system core image library.

Refer to DOS/VS Sort/Merge Programmer's Guide for guidelines in determining the work area space needed.

OUTPUT OF BATCH DICTIONARY EXECUTION

For every batch run, the Dictionary prints (if LOGU=Y) a record of the commands entered and the resulting messages issued. If the operating system is OS/VS, a return code will be indicated. Optionally, a record of internally generated commands may be printed. Individual commands and batch forms can also produce processing reports.

Each message is preceded by a message identification indicating the source of the message and the message type. See Appendix C for an explanation of message identification numbering and a complete list of Dictionary messages with their meanings and suggested user actions.

Some commands specifically request certain types of output. The format and output device may, in some cases, be varied through use of the DEST keyword, as explained in the discussion of each command in Chapter 6.

BATCH INPUT STREAMS

The Dictionary batch input stream may include both card input from batch forms, and commands. Multiple users may also enter commands in the same stream.

If yours is not the only batch input in a card stream, you will always get the default values that were active in the batch immediately before yours in the stream. If you want the installation-defined default values, or if you want to set your own values, you must use the SET commands to change the default values. It is a good idea to always set the default values with the SET commands before entering any other input in your batch job stream.

If you are running a batch job with security enabled, you must always enter the SIGN_ON command ahead of your command or batch form input. The only commands that may precede the SIGN_ON command are the SET and FLUSH commands.

The figure below shows the structure of a typical batch input stream.

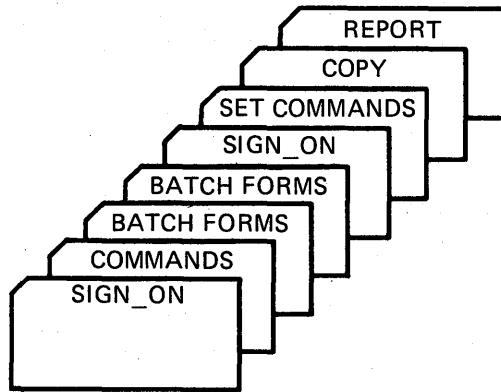


Figure 59. Typical Structure of Dictionary Batch Input

CHAPTER 5. BATCH FORMS PREPARATION

The batch forms input facility furnishes a convenient means for bulk entry of field definitions, segment definitions, and text data. Various types of text data are handled: lines of Description Data, User Data, and PL/I Data. These kinds of Dictionary data lend themselves to bulk input because they are usually entered in large volumes and may be prepared by personnel who do not have access to the Dictionary.

Forms Available

Four preprinted forms designed to simplify data entry are available for batch forms entry. These are:

Field Definition Form; see Figure 60.

Field Definition Form for PL/I Data; see Figure 61.

Segment Definition Form; see Figure 62.

Text Data Form; see Figure 63.

Each form contains appropriately labeled fields for entry of text or other Dictionary data, and instructions for reducing the contents of the form to machine-readable format.

Data may be coded on one or more sheets. You are not limited by the number of lines on one preprinted form. When continuation sheets are needed to fully enter one subject's data, DBDFEND is only coded on the last sheet.

Batch Form Records

A batch form appears as a collection of 80-character card-image records. The first record of each form is the header record. It contains a code which identifies it as the first record of an input form. Information specific to each of the four types of input forms also appears in the header record, as does an identification of the originator of the form. The final record of each input form is an end record, which signals the end of input for that form. Between these beginning and end identifiers, other input data records are coded.

It is important to note that the omission of a header or end record could cause changes intended for one subject to be confused with changes to the Dictionary intended for another subject. For example, an alias intended for one subject could be erroneously added to an earlier subject. Records which represent the continuation of the content of a prior record rely upon the presence of a plus sign (+) in the last coded column of the immediately preceding record.

Mode of Input

The header record contains a field marked MODE. A value of O (old) stipulates that the incoming data for this input form is for an already existing subject. The input will be rejected if the subject is not found in the category. A value of N (new) will, on the other hand, cause the input to be rejected if the subject is present in the data base category.

Batch Form Processing

Batch form input, after data entry, is entered into the normal batch input stream for the Dictionary and can be intermixed with Dictionary commands.

Batch form input is handled in much the same manner as input from the display forms of the interactive display-forms facility. Successive records of the input form are read one at a time from the batch input stream, and the appropriate Dictionary language commands are generated and submitted, internally, to the Dictionary update language processor.

As the batch form input is being processed, a report is produced. This report shows the input data and the results of processing. If an error is encountered in the processing of an input record, an appropriate diagnostic message is printed; processing may or may not be terminated, depending on the severity of the error and the Dictionary FLUSH setting.

Access Considerations

It must be kept in mind that the completed header record is essentially a request to change the contents of the Dictionary. If Dictionary security is active, the submitter of the batch input stream needs to be authorized to make those changes implied by the fields in the header records and other records that specify category and status code. Should this not be the case, then the input form will be flushed and an appropriate error message will be printed. The authorization requirement must be satisfied by entering a suitable SIGN_ON command ahead of the batch form.

Note that the originator designated in the header record is not used for user identification by the Dictionary. This field is printed on the processing report, is not checked in any way, and is not added to the permanent Dictionary data.

Detailed descriptions of the forms and of the records they specify are found in the remainder of this chapter. Note that, in these descriptions, the columns or positions occupied by the field are shown below the name of the field and opposite an explanation of the field.

Examples of filled-out forms and of the reports obtained from processing them are found in Chapter 4 of the DB/DC Data Dictionary Applications Guide.

FIELD DEFINITION FORMS

A Field Definition form is used to collect and record data that defines a field (element). The form allows you to specify alias names for the field, basic parameters (date, type, length), COBOL parameters, and a list of any subfields that occur within the field, with description and language commentary.

A Field Definition form can be used to add a new field definition to the Dictionary or to add data to an existing definition. The form is shown in Figure 60.



DB DC Data Dictionary Field Definition Form

Originator	Date	Punching Instructions	Graphic	Page of
Department		Punch		

Header										Field Subject Name										Defaults																																																										
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
DBDFFLD																																																																														
User Name										Occ										Originator ID																																																										
Status										Language										Occ																																																										

Basic Field Attributes

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
B	F	A												

Description Data

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79											
D	S	C																																																																																							

Aliases

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
A	K	A																																										

Note: Status code for aliases will match that of subject name specified on Header.

PL/I Data

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79											
P	L	I																																																																																							

COBOL Attributes

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71									
C	B	L																																																																													

COBOL "Occurs" Clause

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71									
O	C	C																																																																													

COBOL "Value" Clause

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71									
V	A	L																																																																													

Subfields

Subfield Subject Name										Substart		Subend		Subtype		Length**			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
S	F	D																	

** Not used for COBOL.
** Can only be specified if Mode=N.

End

1	2	3	4	5	6	7
D	B	D	F	F	E	N

Figure 60. Data Dictionary Field Definition Form

The following sections describe the fields and their relative position in the various types of records used for the entry of a field definition.

Header Record

The header record used with the Field Definition form has the following fields:

DBDFFLD

1-7 Specifies that this is a batch input facility entry and that other records follow. The character string DBDFFLD, which is printed on the forms, must be entered.

8-9 Unused

MODE

10 Specifies whether the field definition is to be added to an existing subject definition or used to create a new definition. The valid values for this field are N (new) for a new definition or 0 (old) for an existing definition.

11-18 Unused

Status

19 Specifies the status of the subject field. If a status field is omitted, the status defaults to the value specified in column 74 of this record. If column 74 is blank, the status defaults to the current default value. Valid values are A to T and 0 to 9.

Language

20 Specifies the language portion of the Dictionary name of the subject field. If the language field is omitted, the language defaults to the value specified in column 75 of this record. If column 75 is blank, the language defaults to the current default value. Valid values are A, B, C, J, K, and L.

User Name

21-51 Specifies the user name of the subject field.

Occ

52-54 Specifies the occurrence number portion of the Dictionary name for the subject field. If this field is omitted, the occurrence number defaults to the value specified in columns 77 through 79 of this record. If columns 77 through 79 are blank, the occurrence defaults to the current default value. Valid values are 0 to 255.

Originator ID

55-73 Identifies the originator of the form.

Defaults

74-79 Specifies the defaults to be used while processing this input. These defaults are valid for this header record and remain in effect until the end record is encountered. The values that can be specified in this

field are:

Column 74 Status: A-T, 0-9

Column 75 Language: A, B, C, J, K, L

Column 76 Unused

Columns 77-79 Occurrence Number: 000-255

These defaults are used only if the corresponding fields are left blank in this header record and in any record associated with the header record. If these defaults are not specified, the current defaults will be used.

Basic Field Attribute Record

The basic field attribute record has the following fields:

BFA

1-3 Specifies that this is a basic attribute record. The preprinted character string BFA must be entered.

Type

4 Specifies the type of data to be stored in this field. The valid values for this field are:

- B - Binary digits
- C - Characters
- D - Long floating point
- E - Short floating point
- F - Fixed point binary (word)
- H - Fixed point binary (halfword)
- P - Packed decimal
- X - Hexadecimal digits
- Z - Zoned decimal

Bytes

5-9 Specifies the size of the field in bytes, or bits if Type=B. The value may be from 0 to 32767.

Date

10-15 Specifies a date to be associated with the segment definition. The date is in the form MMDDYY. The range of acceptable dates is from 010100 to 123199.

Description Data Record

The Description Data record can be used to add Description Data to a field definition. This record can be used in place of the Text Data form. The Description Data record has the following fields:

DSC

1-3 Specifies that this is a Description Data record. The preprinted character string DSC must be entered.

Line No.

4-6 Specifies the line number under which the data is to be stored in the Dictionary. The numbers need not be entered in ascending order, nor do they have to be consecutive. You may want to leave numbers for future additions of text. The line numbers must be in the

range 1 to 999. Up to 999 lines of text data can be entered for a given subject. You are not limited by the number of lines on the preprinted form.

Text

7-78 This field contains the text that describes this field definition.

Aliases Record

The aliases record has the following fields:

AKA

1-3 Specifies that this is an aliases record. The preprinted character string AKA must be entered.

4-10 Unused

Code

11 Specifies the language portion of the Dictionary name for the alias. If this field is omitted, it will default to the value specified in column 75 of the header record. If column 75 of the header record is blank, this field will default to the current language default. Valid values are A, B, C, J, K, and L.

Note: The status code for aliases will match that of the subject name specified in the header record.

User Name

12-42 Specifies the user name of the alias for the subject field.

Occ

43-45 Specifies the occurrence number portion of the Dictionary name of the alias. If this field is omitted, it will default to the value specified in columns 77 through 79 of the header record. If columns 77 through 79 of the header record are blank, this field will default to the current occurrence default. Valid values are 0 to 255.

PL/I Data Record

The PL/I data record can be used to add PL/I data to the field definition. This record can be used to enter PL/I data for a field in place of the Text Data form. The PL/I data record has the following fields:

PLI

1-3 Specifies that this is a PL/I data record. The preprinted character string PLI must be entered.

Line No.

4-6 Specifies the line number under which the PL/I data is to be stored in the Dictionary. The numbers need not be entered in ascending order, nor do they have to be consecutive. You may want to leave numbers for future additions of PL/I data. The line numbers must be in the range 1 to 255. Up to 255 lines of PL/I data can be entered for a given subject. You are not limited by the number of lines on the preprinted form.

Type

7 Specifies the type of PL/I data to be entered. The valid values for this field are:

- C - Comment
- N - Narrative
- X - Unsupported attribute

Text

8-79 This field contains the PL/I data to be entered for this field definition.

COBOL Attributes Record

The COBOL attributes record specifies COBOL parameters for the field. This record can be used to enter the COBOL Usage code, Sign, Blank, Justify, and Synch parameters, and the Picture clause. Additionally, this record can specify the Occurs and Value parameters; 5 and 26 bytes are allowed for these, respectively. If this is not enough space, the COBOL occurs clause and COBOL value clause records can be used. The COBOL attributes record has the following fields:

CBL

1-3 Specifies that this is a COBOL attributes record. The preprinted character string CBL must be entered.

Usage

4 Specifies the COBOL Usage code. The valid values are:

- 0 - Computational
- 1 - Computational-1
- 2 - Computational-2
- 3 - Computational-3
- 4 - Computational-4
- D - Display
- I - Index

Sign

5-6 Specifies the COBOL Sign parameter. The valid values are:

- L - Sign leading
- T - Sign trailing
- LS - Sign leading, separate character
- TS - Sign trailing, separate character

Blank

7 Specifies whether the COBOL BLANK WHEN ZERO clause is to be used. To specify the BLANK WHEN ZERO clause, code a B in this field.

Justify

8 Specifies whether the COBOL JUSTIFIED clause is to be used. To specify right justification, code an R in this column.

Synch

9-10 Specifies whether the COBOL SYNCHRONIZE clause is to be used. The valid values are:

- S - Synchronize
- SL - Synchronize left
- SR - Synchronize right

Picture

11-40 Specifies the data for the COBOL PICTURE clause.

Occurs

41-45 Specifies the COBOL Occurs clause. If 5 bytes are not enough space to enter the Occurs clause, you should use the COBOL occurs clause record.

Value

46-71 Specifies the COBOL Value clause. If 26 bytes are not enough space to enter the Value clause, you should use the COBOL Value Clause record described later.

COBOL Occurs Clause Record

The COBOL occurs clause record is used to specify the COBOL Occurs Clause to be stored with the field definition. If the COBOL attributes record was used to specify the Occurs clause, this record should not be used. The COBOL occurs clause record has the following fields:

OCC

1-3 Specifies that this is a COBOL occurs clause record or a continuation of one. The preprinted character string OCC must be entered.

Value

4-63 Specifies the Occurs clause to be stored with the field definition. An Occurs clause has a maximum length of 120 characters. If the clause you enter exceeds 60 characters, place a plus sign (+) in column 64 and continue the clause on the next record in columns 4 through 63.

Cont

64 A plus sign in this field specifies that the occurs data is continued on the next record.

COBOL Value Clause Record

The COBOL value clause record is used to specify the COBOL Value clause to be stored with the field definition. If the COBOL Attributes record was used to specify the Value clause, this record must not be used. The COBOL value clause record has the following fields:

VAL

1-3 Specifies that this is a COBOL value clause record or a continuation of one. The preprinted character string VAL must be entered.

Value

4-63 Specifies the Value clause to be stored with the field definition. A Value clause has a maximum length of 120 characters. If the clause you enter exceeds 60 characters, place a plus sign (+) in column 64 and continue the clause on the next record in columns 4 through 63.

Cont

64 A plus sign in this field specifies that the value data is continued on the next record.

Subfield Record

The subfield record is used to define subfields that are contained within the field defined by this batch form. The subfield record has the following fields:

SFD

1-3 Specifies that this is a subfield record. The preprinted character string SFD must be entered.

4-8 Unused

Mode

9 Specifies whether the subfield already exists in the Dictionary. The valid values for this field are N (new) for a new definition or 0 (old) for an existing definition.

Status

10 Specifies the status of the subfield. If this field is omitted, the default is the status specified in column 74 of the header record. If column 74 of the header record is blank, the current default for status is used. Valid values are A to T and 0 to 9.

Language

11 Specifies the language portion of the Dictionary name for the subfield. If this field is omitted, the default is the value specified in column 75 of the header record. If column 75 of the header record is blank, the current default for language is used. Valid values are A, B, C, J, K, and L.

User Name

12-42 Specifies the user name of the subfield.

Occ

43-45 Specifies the occurrence number portion of the name of the subfield. If this field is omitted, the default is the value in columns 77 through 79 of the header record. If columns 77 through 79 of the header record are blank, the current default value is used. Valid values are 0 to 255.

46 Unused

Use

47 Specifies the type of relationship the subfield has with the field. Any alphabetic character is valid for this field. The user may define the meaning of the value. The following values have a predefined meaning to the Dictionary:

- C - CONTAINS
- D - DEPENDS
- R - REDEFINES
- 6 - RENAMES
- 8 - Conditional names
(COBOL level 88)

Substart

48-53 Specifies the position within the subject field at which the subfield begins. This field is required unless Use=8 has been coded.

Byte

48-52 Specifies the starting byte of the subfield within the subject field. The number can be from 1 to 32767. If USE=8, then substart is automatically set to zero.

Bit

53 Specifies the starting bit within the subfield starting byte. The number can be 1 to 8. This field is only valid for PL/I and assembler language.

Data Type

54 Specifies the data type of the subfield. This field can be specified only if MODE is N (new). The valid values for this field are:

- B - Binary digits
- C - Characters
- D - Long floating point
- E - Short floating point
- F - Fixed point binary (word)
- H - Fixed point binary (halfword)
- P - Packed decimal
- X - Hexadecimal digits
- Z - Zoned decimal

Length

55-59 Specifies the size of the subfield in bytes, or bits if Type=B. This field can be specified only if MODE is N (new). The value can be from 0 to 32767.

End Record

The end record marks the end of a Field Definition form. A report on the field definition can be requested by coding the last field in the end record.

The end record contains the following fields:

DBDFEND

1-7 Specifies that this is the end of this Field Definition form. The preprinted character string DBDFEND must be entered.

Report

8 If an R is coded in this column, a report on the field is produced.

FIELD DEFINITION FORM FOR PL/I DATA

The Field Definition form for PL/I data is used to collect and enter data that defines and describes a PL/I field (or element). The form allows you to specify alias names for the field, basic parameters (date, type, length), PL/I parameters, and a list of any subfields that occur within the field, along with description and PL/I text data.

The Field Definition form can be used to add a new field definition to the Dictionary or to add data to an existing definition. The form is shown in Figure 61.

Originator	Date	Punching Instructions	Graphic Punch	Page	of
Department					

Field Subject Name		Defaults	
Mode (N/O)	Status Language	Status Language	Occ.
1 2 3 4 5 6 7 8 9	10 11 12 13 14 15 16 17 18 19	20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51	52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
DBDFLD			

Basic Field Attributes

Type	Bytes	Date
1 2 3	4 5 6 7 8 9	10 11 12 13 14 15
BFA		

Description Data

Line No.	Type	Text
1 2 3	4 5 6	7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45
DISC		
DISC		
DISC		
DISC		

Aliases

Code	User Name	Occ.
1 2 3	4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45	46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
AKA		
AKA		
AKA		

Note: Status code for aliases will match that of subject name specified on Header.

PL/I Data

Line No.	Type	Text
1 2 3	4 5 6	7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45
PLI		
PLI		
PLI		
PLI		

PL/I Attributes

Format	Option	Mode	Align	Precision		Dimension					
				Sig. Digits	Scale Factor	Lower Boundary 1	Upper Boundary 1	Lower Boundary 2	Upper Boundary 2	Lower Boundary 3	Upper Boundary 3
1 2 3	4 5 6 7	8 9 10 11 12	13 14 15 16	17 18 19 20	21 22 23 24	25 26 27 28 29 30	31 32 33 34 35 36	37 38 39 40 41 42	43 44 45 46 47 48	49 50 51 52 53	54 55 56 57 58 59 60
PAT											

PL/I Picture and Initial

Picture											Initial										
1 2 3	4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33	34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63																			
PIPI																					

Subfields

Subfield Subject Name		Substart	
Mode (N/O)	Status Language	LR	Byte
1 2 3	4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45	46 47 48 49 50 51 52 53	54 55 56 57 58 59
SFD			
SFD			
SFD			
SFD			

*Not used for COBOL
**Can only be specified if Mode=N.

End

IR
1 2 3 4 5 6 7
DBDFEND

Figure 61. Data Dictionary Field Definition Form for PL/I Data

The following sections describe the fields in the various types of records used for a PL/I field definition.

Header Record

The header record for use with the Field Definition Form has the following fields:

DBDFFLD

1-7 Specifies that this is a batch input facility entry and that other records follow. The character string DBDFFLD, which is printed on the forms, must be entered.

8-9 Unused

MODE

10 Specifies whether the field definition is to be added to an existing subject definition or used to create a new definition. The valid values for this field are N (new) for a new definition or O (old) for an existing definition.

11-18 Unused

Status

19 Specifies the status of the subject field. If this field is omitted, the status defaults to the value specified in column 74 of this record. If column 74 is blank, the status defaults to the current default value. Valid values are A to T and 0 to 9.

Language

20 Specifies the language portion of the Dictionary name of the subject field. If this field is omitted, the language defaults to the value specified in column 75 of this record. If column 75 is blank, the language defaults to the current default value. Valid values are A, B, C, J, K, and L.

User Name

21-51 Specifies the user name of the subject field.

Occ

52-54 Specifies the occurrence number portion of the Dictionary name for the subject field. If this field is omitted, the occurrence number defaults to the value specified in columns 77 through 79 of this record. If columns 77 through 79 are blank, the occurrence defaults to the current default value. Valid values are 0 to 255.

Originator ID

55-73 Identifies the originator of the form.

Defaults

74-79 Specifies the defaults to be used while processing this input batch form. These defaults are valid for this header record and remain in effect until the end record is encountered. The values that can be specified in this field are:

Column 74 Status: A-T, 0-9

Column 75 Language: A, B, C, J, K, L

Column 76 Unused

Columns 77-79 Occurrence Number: 0-255

These defaults are used only if the corresponding fields are left blank in this header record and in any record associated with the header record. If these defaults are not specified, the current defaults will be used.

Basic Field Attribute Record

The basic field attribute record has the following fields:

BFA

1-3 Specifies that this is a basic attribute record. The preprinted character string BFA must be entered.

Type

4 Specifies the type of data to be stored in this field. The valid values for this field are:

- B - Binary digits
- C - Characters
- D - Long floating point
- E - Short floating point
- F - Fixed point binary (word)
- H - Fixed point binary (halfword)
- P - Packed decimal
- X - Hexadecimal digits
- Z - Zoned decimal

Bytes

5-9 Specifies the size of the field in bytes, or bits if Type=B. The value can be from 0 to 32767.

Date

10-15 Specifies the date on which batch input will be processed. This is in the form MMDDYY. The range of acceptable dates is 010100 through 123199.

Description Data Record

The description data record can be used to add description data to the field definition. This record can be used in place of the Text Data form. The description data record has the following fields:

DSC

1-3 Specifies that this is a description data record. The preprinted character string DSC must be entered.

Line No.

4-6 Specifies the line number under which the data is to be stored in the Dictionary. The numbers need not be entered in ascending order, nor do they have to be consecutive. You may want to leave numbers for future additions of text. The line numbers must be in the range 1 through 999. Up to 999 lines of text data can be entered for a given subject. You are not limited by

the number of lines on the preprinted form.

Text

7-78 Contains the text that describes this field definition.

Aliases Record

The aliases record has the following fields:

AKA

1-3 Specifies that this is an aliases record. The preprinted character string AKA must be entered.

4-10 Unused

Code

11 Specifies the language portion of the Dictionary name for the alias. If this field is omitted, the default is the value specified in column 75 of the header record. If column 75 of the header record is blank, this field will default to the current language default. Valid values are A, B, C, J, K, and L.

Note: The status code for aliases will match that of the subject name specified in the header record.

User Name

12-42 Specifies the user name of the alias for the subject field.

Occ

43-45 Specifies the occurrence number portion of the Dictionary name of the alias. If this field is omitted, it will default to the value specified in columns 77 through 79 of the header record. If columns 77 through 79 of the header record are blank, this field will default to the current occurrence default. Valid values are 0 to 255.

PL/I Data Record

The PL/I data record can be used to add PL/I data to the field definition. This record can be used in place of the Text Data form. The PL/I data record has the following fields:

PLI

1-3 Specifies that this is a PL/I data record. The preprinted character string PLI must be entered.

Line No.

4-6 Specifies the line number under which the PL/I data is to be stored in the Dictionary. The numbers need not be entered in ascending order nor do they have to be consecutive. You may want to leave numbers for future additions of PL/I data. The line numbers must be in the range 1 through 255. Up to 255 lines of PL/I data can be entered for a given subject. You are not limited by the number of lines on the preprinted form.

Type

- 7 Specifies the type of PL/I data to be entered. The valid values for this field are:
- C - Comment
 - N - Narrative
 - X - Unsupported attribute

Text

- 8-79 This field contains the PL/I data for this field definition.

PL/I Attributes Record

The PL/I attributes record specifies the attributes that an element in a PL/I structure has. They concern the declaration of a PL/I data item. The PL/I attributes record has the following fields:

PAT

- 1-3 Specifies that this is a PL/I attributes record. The preprinted character string PAT must be entered.

Format

- 4-7 Specifies the format of the element. The valid values for the field are:

- BIN - Binary
- BIT - Bit string
- CHAR - Character string
- DEC - Decimal
- PTR - Pointer variable

Option

- 8-12 Specifies the option for the PL/I internal representation. The valid values for the field are:

- FIXED - Fixed point
- FLOAT - Floating point
- VAR - Varying character string

Mode

- 13-16 Specifies whether the element is to be a real or complex number. The valid values are REAL or CPLX, respectively.

Aligned

- 17 Specifies whether the element is to be aligned (Y) or not (N).

Precision

- 18-24 Specifies the precision of an element defined as a decimal number.

Sig. Digits

- 18-20 Specifies the number of significant digits. The valid values are 1 to 255.

Scale Factor

- 21-24 Specifies the scale factor for the numeric element. The valid values are -128 to +127.

Dimensions

25-60 Specifies the bounds for up to three dimension attributes of an element. The lower bounds are optional but, if entered, must be accompanied by an upper bound. Bounds may carry a leading sign (+ or -); a positive sign is assumed if no sign is entered.

The valid values for bounds may be from -32767 to +32767. The lower bound must always be less than or equal to the upper bound.

Lower Bound 1

25-30 Specifies the lower bound of the first dimension attribute of an element.

Upper Bound 1

31-36 Specifies the upper bound of the first dimension attribute of an element.

Lower Bound 2

37-42 Specifies the lower bound of the second dimension attribute of an element.

Upper Bound 2

43-48 Specifies the upper bound of the second dimension attribute of an element.

Lower Bound 3

49-54 Specifies the lower bound of the third dimension attribute of an element.

Upper Bound 3

55-60 Specifies the upper bound of the third dimension attribute of an element.

PL/I Picture and Initial Record

The PL/I picture and initial record is used to specify for an element the PICTURE and/or INITIAL attributes. The PL/I picture and initial record has the following fields:

PPI

1-3 Specifies this is a PL/I picture and initial record. The preprinted character string PPI must be entered.

Picture

4-33 This field is used to specify the PICTURE value of an element. This field allows for a maximum of 30 characters.

Initial

34-63 This field is used to specify the INITIAL value of an element. The value must be enclosed in parentheses. Any embedded quotation mark signs can be specified with a single quotation mark. This field allows for 30 characters maximum, including parentheses.

Subfield Record

The subfield record is used to define subfields that are contained within the field defined by this batch form. The subfield record has the following fields:

SFD

1-3 Specifies that this is a subfield record. The preprinted character string SFD must be entered.

4-8 Unused

Mode

9 Specifies whether the subfield already exists in the Dictionary. The valid values for this field are N (new) for a new definition or 0 (old) for an existing definition.

Status

10 Specifies the status of the subfield. If this field is omitted, the default is the status specified in column 74 of the header record. If column 74 of the header record is blank, the current default for status is used. Valid values are A to T and 0 to 9.

Language

11 Specifies the language portion of the Dictionary name for the subfield. If this field is omitted, the default is the value specified in column 75 of the header record. If column 75 of the header record is blank, the current default for language is used. Valid values are A, B, C, J, K, and L.

User Name

12-42 Specifies the user name of the subfield.

Occ

43-45 Specifies the occurrence number portion of the Dictionary name of the subfield. If this field is omitted, the default is the value in columns 77 through 79 of the header record. If columns 77 through 79 of the header record are blank, the current system default value is used. Valid values are 0 to 255.

46 Unused

Use

47 Specifies the type of relationship the subfield has with the field. Any alphabetic character is valid for this field. The user may define the meaning of the value. The following characters have a predefined meaning to the Dictionary:

- C - CONTAINS
- D - DEPENDS
- R - REDEFINES
- 6 - RENAMES
- 8 - Conditional names
(COBOL level 88)

Substart

48-53 Specifies the position within the subject field at which the subfield begins. This field is required unless Use=8.

Byte

48-52 Specifies the starting byte of the subfield within the subject field. The number can be in the range 1 through 32767. If USE=8, then SUBSTART is automatically set to zero.

Bit

53 Specifies the starting bit within the subfield starting byte. The number can be from 1 through 8. This field is only valid for PL/I and assembler language.

Data Type

54 Specifies the data type of the subfield. This field can be specified only if MODE is coded N (new). The valid values for this field are:

- B - Binary digits
- C - Character
- D - Long floating point
- E - Short floating point
- F - Fixed point binary (word)
- H - Fixed point binary (halfword)
- P - Packed decimal
- X - Hexadecimal digits
- Z - Zoned decimal

Length

55-59 Specifies the size of the subfield in bytes, or bits if Type=B. This field can be specified only if MODE is coded N (new). The value can be from 0 through 32767.

End Record

The end record marks the end of the Field Definition form. A report on the subject field definition can be requested by coding the last field in the end record.

The End record contains the following fields:

DBDFEND

1-7 Specifies that this is the end of this Field Definition form. The preprinted character string DBDFEND must be entered.

Report

8 If an R is coded in this column, a report on the subject field is produced.

SEGMENT DEFINITION FORM

The Segment Definition form is used to record data that defines a segment. The form allows you to specify alias names for the segment, basic physical parameters (date, type, length), and a list of the fields that occur within the segment, with description and PL/I data text.

The Segment Definition form can be used to add a new segment definition to the Dictionary or to add data to an existing definition. The form is shown in Figure 62.



DB/DC Data Dictionary Segment Definition Form

Originator	Date	Punching Instructions	Graphic							Page	of
Department			Punch								

Header

1	2	3	4	5	6	7	8	9	Mode (NO)	11	12	13	14	15	16	17	18	19	Status	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
Segment Subject Name										Segment Name										Occ.										Originator ID										Defaults																																							
										DBDFSEIG																																																																					

Basic Segment Attributes

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Max Bytes				Min Bytes				No. Flds				Data				Align						
BSA																						

Description Data

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
Line No.										Text																																																																					
DSC																																																																															
DSC																																																																															
DSC																																																																															
DSC																																																																															
DSC																																																																															
DSC																																																																															
DSC																																																																															

Aliases

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
Code										User Name										Occ.																								
AKA																																												
AKA																																												
AKA																																												

Note: Status code for aliases will match that of subject name specified on Header.

PL/I Data

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
Line No.										Text																																																																					
PLI																																																																															
PLI																																																																															
PLI																																																																															
PLI																																																																															
PLI																																																																															

Related Fields

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
Mode (NO)										Subject Name										Start										Length**																											
										User Name										Occ.										Byte										Data Type																	
RFD																																																									
RFD																																																									
RFD																																																									
RFD																																																									
RFD																																																									
RFD																																																									
RFD																																																									
RFD																																																									
RFD																																																									

*Not used for COBOL.
**Can only be specified if Mode=N.

End

1	2	3	4	5	6	7	8
DBDFEND							

Figure 62. Data Dictionary Segment Form

The following sections describe the fields in the records that define a segment.

Header Record

The header record for use with the Segment Definition form has the following fields:

DBDFSEG

1-7 Specifies that this is a batch input facility entry and that other records follow. The preprinted character string DBDFSEG must be entered.

8-9 Unused

Mode

10 Specifies whether the segment definition is to be added to an existing subject definition or used to create a new definition. The valid values for this field are N (new) for a new definition or O (old) for an existing definition.

11-18 Unused

Status

19 Specifies the status portion of the Dictionary name for the subject segment. If this field is omitted, the status default is the value specified in column 74 of this record. If column 74 is blank, the status defaults to the current default value. Valid values are A to T and 0 to 9.

Language

20 Specifies the language portion of the Dictionary name for the subject segment. If this field is omitted, the default is the value specified in column 75 of this record. If column 75 is blank, the language defaults to the current default value. Valid values are A, B, C, J, K, and L.

Segment Name

21-51 Specifies the user name portion of the Dictionary name for the subject segment.

Occ

52-54 Specifies the occurrence number portion of the Dictionary name for the subject segment. If this field is omitted, the occurrence number defaults to the value specified in columns 77 through 79 of this record. If columns 77 through 79 are blank, the occurrence defaults to the current default value. Valid values are 0 to 255.

Originator ID

55-73 This field identifies the originator of the form.

Defaults

74-79 Specifies the defaults to be used while processing this input. These defaults are valid for this header record and remain in effect until the end record is encountered. The values that can be specified in this field are:

Column 74 Status A-T, 0-9

Column 75 Language A, B, C, J, K, L

Column 76 Unused

Columns 77-79 Occurrence Number 0-255

These defaults are used only if the corresponding fields are left blank in this header record and in any record associated with the header record. If these defaults are not specified, the current defaults will be used.

Basic Segment Attributes Record

The basic segment attributes record defines the physical attributes of the segment being defined. The basic segment attributes record has the following fields:

BSA

1-3 Specifies that this is a basic segment attributes record. The preprinted character string BSA must be entered.

Max Bytes

4-8 This field specifies the maximum length of a variable-length segment or the number of bytes in a fixed-length segment. The value can be from 0 through 32767.

Min Bytes

9-13 Specifies the minimum length of a variable-length segment. The value can be from 0 through 32767.

No. Flds

14-16 Specifies the number of fields in the segment. The value can be from 0 through 255.

Date

17-22 Specifies a date to be associated with the segment definition. The date is in the form MMDDYY. The range of acceptable dates is from 010100 to 123199.

Aligned

23 Specifies desired boundary alignment. The valid values are Y (yes) if alignment is desired, or N, the default, if it isn't desired.

Description Data Record

The description data record can be used to add description data to the segment definition. This record can be used in place of the Text Data form. The description data record has the following fields:

DSC

1-3 Specifies that this is a description data record. The preprinted character string DSC must be entered.

Line No.

4-6 Specifies the line number under which the data is to be stored in the Dictionary. The numbers need not be entered in ascending order, nor do they have to be consecutive. You may want to leave numbers for future

additions of text. The line numbers must be from 1 through 999. Up to 999 lines of text data can be entered for a given subject. You are not limited by the number of lines on the preprinted form.

Text

7-78 This field contains the text that describes this segment definition.

Aliases Record

The aliases record has the following fields:

AKA

1-3 Specifies that this is an aliases record. The preprinted character string AKA must be entered.

4-10 Unused

Code

11 Specifies the language portion of the Dictionary name for the alias being defined. If this field is omitted, the default in column 75 of the header record is used. If column 75 of the header record is blank, the default is the current language default. Valid values are A, B, C, J, K, and L.

Note: The status code for aliases will match that of the subject name specified in the header record.

User Name

12-42 Specifies the user name of the alias for the subject segment.

Occ

43-45 Specifies the occurrence number portion of the Dictionary name for the alias being defined. If this field is omitted, the default is the value specified in columns 77 through 79 of the header record. If columns 77 through 79 of the header record are blank, the default is the current default. Valid values are 0 to 255.

PL/I Data Record

The PL/I data record can be used to add PL/I data to the segment definition. This record can be used in place of the Text Data form. The PL/I data record has the following fields:

PLI

1-3 Specifies that this is a PL/I Data record. The preprinted character string PLI must be entered.

Line No.

4-6 Specifies the line number under which the PL/I data is to be stored in the Dictionary. The numbers need not be entered in ascending order nor do they have to be consecutive. You may want to leave numbers for future additions of PL/I data. The line numbers must be from 1 through 255. Up to 255 lines of PL/I data can be entered for a given subject. You are not limited by the number of lines on the preprinted form.

Type

7 Specifies the type of PL/I data to be entered. The valid values for this field are:

- C - Comment
- N - Narrative
- X - Unsupported attribute

Text

8-79 This field contains the PL/I data for this segment definition.

Related Field Record

The related field record specifies fields that occur within the subject segment. One record must be coded for each field to be defined. The related field record contains the following fields:

RFD

1-3 Specifies that this is a Related Field record. The preprinted character string RFD must be entered.

4-8 Unused

Mode

9 Specifies whether the related field already exists in the Dictionary. The valid values for this field are N (new) for a new definition or 0 (old) for an existing definition.

Status

10 Specifies the status of the related field. If this field is omitted, the default is the status specified in column 74 of the header record. If column 74 of the header record is blank, the current default for status is used. Valid values are A to T and 0 to 9.

Language

11 Specifies the code portion of the Dictionary name for the related field to be added. If this field is omitted, the default is the value specified in column 75 of the header record. If column 75 of the header record is blank, the current default for language is used. Valid values are A, B, C, J, K, and L.

User Name

12-42 Specifies the user name of the related field.

Occ

43-45 Specifies the occurrence number portion of the Dictionary name for the related field to be added. If this field is omitted, the default is the value in columns 77 through 79 of the header record. If columns 77 through 79 of the header record are blank, the current default value is used. Valid values are 0 to 255.

Start

46-51 This required field specifies the position within the segment at which this field begins.

Byte

46-50 Specifies the starting byte at which the field begins. The number can be from 1 through 32767.

Bit

51 Specifies the starting bit within the starting byte at which the begins. The number can be from 1 through 8. This field is only valid for PL/I and assembler language. It is not valid for COBOL.

Sequence

52 Specifies whether or not this is a sequence field. The valid values are:

blank - not used in a DBD

A - Ascending sequence

D - Descending sequence

U - Sequence field with unique values

M - Sequence field with duplicate values

G - Not a sequence field, but generate a FIELD statement for DBD_OUT

Data Type

53 Specifies the type of data to be stored in this field. This field can be used only if MODE is coded N (new). The valid values for this field are:

B - Binary digits
C - Characters
D - Long floating point
E - Short floating point
F - Fixed-point binary (word)
H - Fixed-point binary (halfword)
P - Packed decimal
X - Hexadecimal digits
Z - Zoned decimal

Length

54-58 Specifies the length of the field in bytes, or bits if Type=B. This field can be used only if Mode is coded N (new). The value can be from 1 through 32767.

End Record

The end record marks the end of a Segment Definition form. A report on the segment definition can be requested by coding the last field in the end record.

The End record contains the following fields:

DBDFEND

1-7 Specifies that this is the end of this Segment Definition form. The preprinted character string DBDFEND must be entered.

REPORT

8 If an R is coded in this column, a report of the subject segment will be produced.

TEXT DATA FORM

The Text Data form is used to record Description Data, PL/I Data, or User Data text that is to be added to a subject definition. The form is shown in Figure 63.



DB/DC Data Dictionary Text-Data Form

Originator	Date	Punching Instructions	Graphic Punch	Page of
Department				

Header	Subject Name	Defaults
Type (DSC/PLI/USR)	User Data No. Mode (N/D)	Status Language
Subject Category	User Name	Occ
Originator ID	Status Language	DB Type
DBDF		

Input Text	Text
Line No.	Type
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	
35	
36	
37	
38	
39	
40	
41	
42	
43	
44	
45	
46	
47	
48	
49	
50	
51	
52	
53	
54	
55	
56	
57	
58	
59	
60	
61	
62	
63	
64	
65	
66	
67	
68	
69	
70	
71	
72	
73	
74	
75	
76	
77	

End	DB/DC
Line No.	Type
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	
35	
36	
37	
38	
39	
40	
41	
42	
43	
44	
45	
46	
47	
48	
49	
50	
51	
52	
53	
54	
55	
56	
57	
58	
59	
60	
61	
62	
63	
64	
65	
66	
67	
68	
69	
70	
71	
72	
73	
74	
75	
76	
77	

Header	Subject Name	Defaults
Type (DSC/PLI/USR)	User Data No. Mode (N/D)	Status Language
Subject Category	User Name	Occ
Originator ID	Status Language	DB Type
DBDF		

Input Text	Text
Line No.	Type
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	
35	
36	
37	
38	
39	
40	
41	
42	
43	
44	
45	
46	
47	
48	
49	
50	
51	
52	
53	
54	
55	
56	
57	
58	
59	
60	
61	
62	
63	
64	
65	
66	
67	
68	
69	
70	
71	
72	
73	
74	
75	
76	
77	

End	DB/DC
Line No.	Type
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	
35	
36	
37	
38	
39	
40	
41	
42	
43	
44	
45	
46	
47	
48	
49	
50	
51	
52	
53	
54	
55	
56	
57	
58	
59	
60	
61	
62	
63	
64	
65	
66	
67	
68	
69	
70	
71	
72	
73	
74	
75	
76	
77	

Figure 63. Data Dictionary Text Data Form

The following sections describe the fields in the three types of records used for text data entry.

Header Record

The header record for use with the Text Data form has the following fields:

DBDF

1-4 Specifies that this is a batch input facility entry and that other records follow. The characters DBDF, which are printed on the forms, must be entered.

Type

5-7 Specifies the type of information to follow. Only one type of data can be entered with a single Text Data form. The valid values for this field are:

DSC Specifies that description data follows.

PLI Specifies that PL/I data follows.

USR Specifies that user data text follows.

8 Unused

User Data Number

9 Specifies the number of the user data segment to which the text applies. This field is used only if USR is coded in TYPE. The valid value is 1, 2, 3, 4, or 5.

Mode

10 Specifies whether the text data is to be added to an existing subject definition or used to create a new definition. The valid values for this field are N (new) for a new definition or 0 (old) for an existing definition.

Subject Category

11-18 Specifies the category of the subject to which the text is to be added. The categories and valid values for this field are shown in Figure 8.

Only the ELEMENT and SEGMENT categories are valid if the input text is PL/I data.

Status

19 Specifies the status portion of the Dictionary name for the subject to which this text is to be added. If this field is omitted, the status default is the value specified in column 74 of this record. If column 74 is blank, the status defaults to the current default value. Valid values are A to T, 0 to 9 and *.

Code

20 Specifies the code qualifier of the Dictionary name for the subject to which this text is to be added. If this field is omitted and the category is ELEMENT or SEGMENT the code defaults to the value specified in column 75 of this record. If column 75 is blank, the code defaults to the current language default value.

If this field is omitted and the subject category is DATABASE, the code defaults to the value specified in column 76 of this record. If column 76 is blank, the code defaults to the current default value for the

data base type.

If this field is omitted and the category is a standard category other than ELEMENT, SEGMENT, or DATABASE, the code defaults to the correct value for the subject. For example, if the category is TRANSACTION, the code would default to T. If the category is PGM, the code would default to G.

If the category is an installation-defined category, no language code value may be specified.

User Name

21-51 Specifies the user name portion of the Dictionary name for the subject to which the text data is to be added.

Occ

52-54 Specifies the occurrence number portion of the Dictionary name for the subject to which the text data is to be added. If this field is omitted, the occurrence number defaults to the value specified in columns 77 through 79 of this record. If columns 77 through 79 are blank, the occurrence defaults to the current default value. Valid values are 0 to 255.

Originator ID

55-73 This field identifies the originator of the form.

Defaults

74-79 These fields specify the defaults to be used while processing this input. These defaults are valid for this header record and remain in effect until the end record is encountered. The values that can be specified in this field are:

Column 74 Status A-T, 0-9, *

Column 75 Language A, B, C, J, K, L

Column 76 DB Code P, L, F, X, Y

Columns 77-79 Occurrence Number: 0-255

These defaults are used only if the corresponding fields are left blank in this header record and in any record associated with the header record. If these defaults are not specified, the current defaults will be used, except as noted in the discussion of the language field above.

Input Text Record

The input text record has the following fields:

Line No.

1-3 Specifies the line number under which the data is to be stored in the Dictionary. The numbers need not be entered in ascending order nor do they have to be consecutive. You may want to leave numbers for future additions of text. The line numbers for description data and user data must be in the range 1 through 999. The line numbers for PL/I data must be in the range 1 through 255. You are not limited by the number of lines on the preprinted form.

Type

- 4 Specifies the type of PL/I data to be entered if PLI was specified on the header record. The value can be:
- C - Comment
 - N - Narrative
 - X - Unsupported attribute

If the input text is description or user data, this field should be left blank.

Text

- 5-76 This field contains the text to be associated with the subject specified in the header record. To enter a full 80-byte record with this form, enter bytes 1 through 72 in columns 5 through 76, place a plus sign (+) in column 77, and put the final 8 bytes in the next record starting in column 5. When a continuation is indicated, the line number must be repeated in columns 1 through 3 of the continuation record. Use only columns 5 through 76 for description or PL/I data.
- 77 A plus sign in this field specifies that the text is continued on the next record.

End Record

The end record has the following fields:

DBDFEND

- 1-7 Specifies that this is the end of this Text Data form. The preprinted character string DBDFEND must be entered.

Report

- 8 Specifies whether you want a report on the subject to which the text was added, and, if so, what type of report. If A (for All) is coded, a report on subject type and subject name specified in the header record is printed. If T (for Text) is coded, a report on only the text data associated with the subject type and subject name is printed. The default is no report.

CHAPTER 6. COMMANDS

This chapter contains detailed descriptions of all Dictionary commands including any options, default values, and restrictions that apply. More information about the commands and examples of how to use them may be found in the DB/DC Data Dictionary Applications Guide. The names of the commands, and the function they perform are summarized below. Assuming a full function Dictionary is being used, all of these commands are available to all users in batch mode. Batch commands are entered in card-image form. In batch operation, the use of the commands is almost identical for DOS/VS and OS/VS systems. Minor differences (in keyword values, for example) are pointed out in the description of each command. All messages that can be received when working with the command language are presented in Appendix C.

Online users may not use the DBD_IN, PSB_IN, PLI_IN, and COBOL_IN commands which are available in batch mode only. IMS/VS online users may not use the CKPT and SETCKPT commands. See the section on the COMMAND form in Chapter 3 for details about using commands online.

Command	Function
Processing Control Functions	
SIGN_ON	Identifies a user to the Dictionary.
CKPT	Requests a DL/I checkpoint.
SETSTAT, SETOCR, SETDBTP,	
SETLANG	Changes subject name qualifier defaults for current batch run or terminal session.
SETCKPT	Sets a DL/I checkpoint frequency for batch.
SETLOGU, SETLOGI	Sets the logging option for user or Dictionary generated commands.
DEFAULTS	Obtains the current status of the Dictionary control defaults.
FLUSH	Specifies whether or not processing should continue if an error is encountered. Batch only.

System Description Processing Functions

DBD_IN (DI)	Reads DBDs from a DBD library and creates Dictionary entries for data bases, segments, elements, and relationships. Batch only.
PSB_IN (PI)	Reads PSBs from a PSB library and creates Dictionary entries for PSBs, PCBs, programs, and relationships. Batch only.
DBD_OUT (DO)	Retrieves related entries describing a data base structure and produces source statements for the DBD generation process in the host DL/I system.
PSB_OUT (PO)	Retrieves related entries describing a PSB and produces source statements for the PSB generation process in the host DL/I system.
DDT_OUT	Produces a set of Data Definition Table (DDT) source statements suitable for editing into the source input stream for GIS DDT libraries.
STAGE_1_OUT (S10)	Produces Stage 1 System Definition information for IMS/VS system definition.

Update and Copy Functions

ADD (A)	Adds subjects and subject data.
ADD_RELATIONSHIP (AR)	Adds relationships and relationship data.
CHANGE_IN (CI)	Changes subject data (and deletes individual keyword values).
CHANGE_NAME (CN)	Changes subject names.
CHANGE_RELATIONSHIP_DATA (CRD)	Changes relationship data (and deletes individual keyword values).
DELETE (DEL)	Deletes subjects and complete associated definitions.
DELETE_DATA (DD)	Deletes subject data (by Dictionary segment names).
DELETE_RELATIONSHIP (DR)	Deletes relationships and all associated relationship data.
DELETE_RELATIONSHIP_DATA (DRD)	Deletes relationship data (by Dictionary segment name).
DELETE_STRUCTURE	Deletes a subject name, related subject names and definitions (from a specified level to a specified level).
EXTEND_RELATIONSHIP (XR)	Relates fields of a segment or subfields of a field to a given subject.

RELOCATE (RL)	Changes hierarchical order of segments related to a data base or PCB.
COPY	Copies a subject name, related subject names and definitions, changing subject names as specified.
 Scan and Report Functions	
REPORT (RPT)	Retrieves requested information from the Dictionary: glossary reports (lists), details for specified subjects, or batch display form equivalents.
SCAN (SC)	Finds subjects passing specified criteria and produces requested reports.
PUNCH	Punches user data text.
EXECUTE (EXEC)	Executes a user program to retrieve data from the Dictionary data bases.

Programming Language Data Structure Processing Functions

COBOL_IN (CBI)	Reads member(s) of a COBOL copy library or a sequential data set and creates appropriate Dictionary entries for segments, elements, and relationships. Batch only.
PLI_IN	Reads member(s) of a PL/I %INCLUDE library or a sequential data set and creates appropriate Dictionary entries for segments, elements, and relationships. Batch only.
STRUCTURES_OUT (SO)	Retrieves segment and element entries and creates programming language data structures suitable for assembler language, PL/I, or COBOL programs.
RECALCULATE_SEGMENT (RS)	Recalculates length (when it has changed) and starting positions in segments.

COMMAND CONVENTIONS

The conventions that follow apply to all commands regardless of whether they are entered in a batch input stream, on the COMMAND form, or from CMNDMOD.

THE CHARACTER SET USED IN COMMANDS

The Dictionary recognizes as a character any of the 8-bit configurations of the extended binary coded decimal interchange code (EBCDIC).

In some operands, you can specify only numeric values; that is, only the digits 0 through 9. In other operands, only alphabetic entries are accepted; that is, the 26 letters of the English alphabet, A through Z. When an operand is said to accept alphameric entries, it means that you can enter any alphabetic or numeric character, or the following: + (dollar sign), # (pound sign), @ (at sign), / (slash), - (hyphen), _ (underscore or underline), or . (period).

Special Characters

Special characters are those that can delimit a command. Besides the blank, these characters are: ((left parenthesis),) (right parenthesis), = (equal sign), ' (single quotation mark), or , (comma).

ENTERING COMMANDS

Commands are entered as 79-character records. Position 80 in the record is ignored. Each command must start on a new line or in a new card image. (But a command verb can be preceded by one or more blanks.)

The plus sign (+) is used to indicate that the command is continued on the next input record. The plus sign indicates where the continuation of the command will start. The first column of the following card will overlay the plus sign. If you want a blank between the end of a command on one line and its continuation on the next, you must provide it before the plus sign or at the beginning of the continuation line. A command may be broken at any point, but if it is broken between operands, there must be a blank at the end of one line or the beginning of the next. Never begin a continuation of a command with an asterisk in column 1; if you do, the continuation is interpreted as a comment and an error message is issued.

The length of a command, not counting the continuation characters, must not exceed 3,840 characters. Blank input lines between parts of a command are ignored. They do not add to the length of the command on the COMMAND display screen.

If a character string contains special characters, including blank, it must be enclosed in single quotation marks (''). For example, the PL/I structure INITIAL('ABC') would appear in a command as PLIINIT=('('ABC')'). Any single quotation marks that appear within the quoted string must be replaced with a pair of single quotation marks. Exceptions are comments within the comment stream, which do not need quotation marks for special characters, and the VALUE keyword, where quotation marks are required for alphameric entries or displayed number values.

A command must end with a semicolon (;) if it is followed by a comment in the same record or if it includes a semicolon or a plus sign in a character string. Otherwise, the absence of the continuation sign (+) is enough to mark the end of the command.

Comments can be entered between commands but they must not interrupt a command. A comment can begin on the same line with the end of a command, if the command ends with a semicolon; or it can begin on a new line with an asterisk (*) in column 1. Each new line of the comment must also begin with an asterisk in column 1. The continuation sign (+) is reserved for commands; it must not be used at the end of a line of a comment.

GENERAL COMMAND FORMAT

Each command is presented in a general format like that in Figure 64. The example does not represent a real command; it merely illustrates some common operands and important rules for entering commands. Examples of real commands may be found in Figure 66.

```
COMMAND_VERB category subjectname [keyword=value] [...] +
[id=value] [...] [DEST={P|L|T}];comment
*comment continued
```

Figure 64. General Command Format

Operands shown in uppercase letters represent keywords that have a predefined meaning in the Dictionary. These operands must be spelled exactly as shown. Abbreviations for operands and values are not shown in the command syntax. See the descriptions that follow the syntax examples for abbreviations and options.

Operands shown in lowercase letters represent information you should supply.

The parts of the general command format are explained below:

COMMAND_VERB

The full name of the command or its abbreviation. The abbreviation will appear along with the command above its description. The verb is entered in uppercase letters. Multiple-word commands are connected by underscores.

category

This operand specifies a Dictionary subject category. Use the full category name or a valid abbreviation.

subjectname

This operand may occur more than once in a single command. It is used in conjunction with the category field and names a subject in that category. Refer to the specific command for the meaning and use of this operand. See Figure 2 in Chapter 1 for the rules for forming subject names.

keyword=value

This operand specifies subject or relationship data to be added or related to the subject definition. No blanks should appear in the keyword=value expression. Enter as many keywords as are required. If the same keyword is specified more than once in a command, a warning message may be issued. The Dictionary will use only the last value specified for the duplicated keyword. If you specify more than one value with a keyword rd. If you specify more than one value with a keyword, enter the values the way they are formatted in the description of the individual command. Two formats are possible.

- keyword=(value1,value2,...)
- keyword=((value1),(value2),(...))

See Appendixes A and B to find keywords, and their valid values. Both appendixes are organized by standard category. Appendix A contains keywords for subject data. Appendix B contains keywords for relationship data. To find out more about keywords see "Keywords in Commands" below.

...

The ellipsis indicates that multiple entries of the type immediately preceding the ellipsis are allowed.

+ The plus sign is used to show that the command is continued on the next line.

id=value
This operand is required with some keywords. No blanks should appear in the id=value expression. See "Keywords in Commands" below for the details of its use.

DEST=P|L|T
An optional operand that specifies the desired output device. See individual commands for options and defaults. Valid values are:

L	Line printer
P	Punch
T	Terminal

If the operand is omitted, the default destination (designated by underlining in the command syntax) is assumed.

; Specifies the end of a command.

comment
An optional freeform comment. It can be entered on the same line as the command if separated from it by a semicolon, or on a separate line if an asterisk is placed in column 1. A plus sign cannot be used to continue a comment.

***comment continued**
An asterisk is required for each new line of comment. It must be coded in column 1.

KEYWORDS IN COMMANDS

Keywords allow you to enter relationships and attributes into the Dictionary with commands. If the keyword names an attribute, the value following the keyword will be assigned to that attribute for the subject specified. A relationship keyword describes the kind of relationship that exists between two subjects in a command. The sections below describe six types of keywords.

Standard Keywords

Commands exist to allow you to modify and add to subjects and relationships in the Dictionary. The standard keywords, referred to simply as keywords, allow you to change or store data in the Dictionary. The keywords themselves are usually attribute names. They specify the attribute to be changed or added and the value it is to receive. They always occur in the form keyword=value.

Some commands require certain keywords. When required, keywords are explained in the description of the commands. All the standard keywords and their valid values are contained in Appendixes A and B at the back of this book.

Installation-defined Keywords

These keywords perform the same function for subjects in installation-defined categories that standard keywords do for subjects in standard categories. They are listed, along with their valid values, in the Guide reports for your installation. See Chapter 12 of the DB/DC Data Dictionary Applications Guide for more information about the Guide report.

Some installation-defined keywords may be specified more than once in the same command. This capability is indicated in the Guide report by the repeat factor, which, if greater than one, indicates that multiple values may be specified. These keywords, called repeating attributes, must be specified in the form: keyword(instance)=value, where the value of instance must be less than or equal to the repeat factor.

Standard Relationship Keywords

Relationships between two subjects can be interpreted in more than one way. To indicate what those relationships are, relationship keywords are used. For standard categories these relationships are defined by the Dictionary. The keywords that describe relationships between standard categories are also defined and fixed by the Dictionary. These keywords have been chosen to explain the relationship they create. For example, the relationship keyword **CONTAINS** in the statement **ELEMENT (elementnameA) CONTAINS ELEMENT (elementnameB)** clearly describes the relationship between the superior element (A) and the subordinate element (B). The relationship described is unique and would be different if the positions of the element names were reversed.

The keyword **WITH** differs from the other standard keywords in that the order of the related subjects does not matter. For example, **DATABASE (databasename) WITH SEGMENT (segmentname)** has the same meaning as **SEGMENT (segmentname) WITH DATABASE (databasename)**.

Relationship keywords are required for all commands which process relationships. In the description of every command that uses one there is an explanation of the use of required and optional relationship keywords. Standard relationship keywords appear in uppercase bold letters in commands.

Installation-defined Relationship Keywords

There are no standard keywords provided with the Dictionary to describe relationships between subjects in installation-defined categories (other than alias relationships). The Dictionary therefore provides a means for your installation to define relationship keywords for subjects in the categories it has developed.

Relationships between installation-defined categories and standard or other installation-defined categories are described and defined by **forward and inverse names**. These names describe the relationship in terms of the first category specified. For example, consider the categories "SON" and "FATHER." A subject in the SON category, if it were to be related to another in the FATHER category, would have the forward name of "SON_OF." It would have the inverse name of "FATHER_OF." The forward name describes the relationship the first subject has with the second, and the inverse name describes the same relationship from the second subject's point of view. Therefore, if the first subject were in the FATHER category, the relationship keyword used would be **FATHER_OF** to describe the same relationship to the subject in the SON category. Notice that the forward and inverse names for the first subject are the inverse and forward names for the second.

The Guide report describes the relationship keywords and other information your installation has defined for use with installation-defined categories. Refer to Chapter 12 of the DB/DC Data Dictionary Applications Guide for more information about Guide reports.

Id Keywords

Some keywords require additional information be supplied before the values they specify are usable. Because these keywords may have more than one value stored for the same subject, they need some way of identifying which value is being referenced. The information must be supplied by specifying an additional keyword called an **id**. These id keywords are described, where applicable, in Appendixes A and B. If a keyword requires an id, both the id, and the keyword must be in the same command. Keywords which require different ids may be specified in the same command. Keywords which require the same id but with different values must be processed in separate commands.

Sequence Attribute Keywords

Some installation-defined relationship keywords define relationships that can occur more than once between the same two subjects. To identify which relationship occurrence is being referenced, you must always specify the **sequence attribute** along with a value. If you don't know the name of the sequence attribute keyword, you can use the general-purpose keyword *SEQ with the value you need. The Guide report describes the sequence attribute (if any) for an installation-defined relationship-type.

SUBJECT NAME MEMORY (DITTO)

In working with the Dictionary, you may frequently enter a series of commands dealing with the same subject. To help you save time and avoid keying errors, the Dictionary includes a subject name memory. This "memory" saves the last subject name used in commands for each of the standard subject categories. The current names in the subject name memory are kept only for one batch run or one terminal session. (IMS/VS online users should, however, clear their scratch pad area (SPA) with an /EXIT command at the beginning and end of a session to avoid possible problems.)

The subject name memory does not apply to installation-defined categories. In addition, there are some commands to which the subject name memory does not apply: COBOL_IN, SCAN, PLI_IN, (none of which requires a subject name in the command), DELETE, DELETE_STRUCTURE, DBD_IN, DBD_OUT, PSB_IN, PSB_OUT, STAGE_1_OUT, and STRUCTURES_OUT.

The memory stores both a subject A and a subject B from the update commands that have two subject names in two different categories. If the subject A and subject B categories are the same, only subject A will be kept. The memory does not store the third subject, subject C, that sometimes appears in an ADD_RELATIONSHIP or RELOCATE command.

No subject name is kept in the memory from a command containing a syntax error. For the CHANGE_NAME and COPY commands, only subject B will be kept in the memory.

To recall a subject name from the subject name memory in one of the appropriate commands, enter the one- or three-letter abbreviation of the subject category followed by a ditto mark ("), with no space between the abbreviated category name and the ditto. The abbreviated category name and ditto mark replace the category and subject name that otherwise appear in the command.

You can use the ditto for both subject A and subject B if the two subjects are in different categories and the last name used for each category is the one your command refers to. If subject A and subject B are in the same category, you can use the ditto for either subject, but not for both.

If you use the ditto mark inappropriately (for example, in one of the commands listed above, or to refer to a category for which no subject name is currently in the memory), you will

```

* COMMENTS START WITH AN ASTERISK IN COLUMN 1
* AND CAN BE CONTINUED TO THE FOLLOWING LINES
*
*
ADD DTE FIELD1 TEXTSEQ=3 TEXT='THIS EXAMPLE SH+
OWS HOW A WORD MAY BE SPLIT US+
ING A PLUS SIGN'; NEEDS SEMICOLON TO ADD THIS COMMENT

ADD DTE 'FIELD(1)';          SINCE THE USER NAME CONTAINS
*                           SPECIAL CHARACTERS, IT MUST
*                           BE ENCLOSED IN QUOTES.

DELETE_DATA DTE FIELDA DATA=((DTEATR), (DTEDESCR));
* THE ABOVE COMMAND SHOWS THE USE OF PARENTHESES IN
* A DELETE_DATA COMMAND, AND THE USE OF ONE OR MORE
* BLANKS BETWEEN OPERANDS, BUT NOT EITHER SIDE OF THE =

AR DTE FIELDA CONTAINS DTE FIELDB SUBSTART=3;
* THIS COMMAND SHOWS A REQUIRED KEYWORD 'SUBSTART'
* WHICH IS AN ID KEYWORD
*
*
AR DTE FIELDA CONTAINS DTE FIELDB SUBSTART=3 USE=A;
* AGAIN THE ID KEYWORD IS REQUIRED WITH THE
* OTHER RELATIONSHIP DATA KEYWORD
*
*
CRD DTE FIELDA WITH DTE FIELDB FROM SUBSTART=3 USE=A+
                                TO SUBSTART=3 USE=B;
* THE ID KEYWORD IS REQUIRED ON BOTH SIDES OF THE
* CHANGE_RELATIONSHIP_DATA COMMAND

* THE FOLLOWING COMMANDS SHOW THE USE OF DITTO
* WITH ONE AND 3 CHARACTER CATEGORY ABBREVIATIONS
* THEY ALSO ILLUSTRATE THAT THE SUBJECT MEMORY
* IS FOR EACH STANDARD CATEGORY
A SEG SEGAA;
A DTE FIELDX;
A DTE" TYPE=C;
A E" BYTES *;
A S" ALIGN=Y;
A DTE" DATE=030380;

```

Figure 65. Examples of Command Syntax

receive an error message.

ADD COMMAND

This command can be used to:

- Add a new subject name to a specified category
- Add a new subject name and subject data to a specified category
- Add subject data to an existing subject name

The data that can be added varies according to the category and is independent of any relationships the subject participates in. Subjects in all categories can have description and user data entries associated with them.

The general form of this command is:

```
ADD category subjectname [keyword=value] [...] +  
[id=value] [...];
```

The parts of the command are explained below:

ADD

A

The long and short command verbs.

category

A required operand that specifies the category to which information is to be added.

subjectname

A required operand that specifies the subject name to be added and/or to have information added to it.

id=value

This operand specifies the id keyword. It is required if shown after a keyword in Appendix A.

ADD RELATIONSHIP COMMAND

This command can be used to:

- Enter a relationship between two subjects
- Add relationship data to an existing relationship entry
- Add a relationship entry with relationship data
- Create aliases
- While accomplishing any of the above, add one or both subject names specified

The `ADD_RELATIONSHIP` command varies somewhat according to whether relationship data is included and which subject categories are involved.

Restrictions on the Status of Subjects

The subjects used in this command must meet the following tests:

- If an alias is being added, both subject names must be of the same status.

- For other uses of the command, both subjects must be of the same status or the subject lower in the Dictionary hierarchy must have a production status.

The Dictionary hierarchy for subjects in standard categories is described in Chapter 5 of the DB/DC Data Dictionary Applications Guide. For subjects in installation-defined categories, check the Guide reports. The Guide report for the category will indicate which subject (if either) is the superior subject. If the relationship is DIRECTED, the lefthand category, specified in the relationship-type Guide report, is the higher category in the relationship.

Standard Relationships

A standard relationship is a relationship between subjects in standard categories. It may include relationship data. Relationships that are not standard are treated individually in the sections that follow this one. To add a standard relationship, the command takes the following form:

```
ADD_RELATIONSHIP categoryA subjectnameA WITH +
categoryB subjectnameB [keyword=value][...];
```

The parts of the command are described below:

ADD_RELATIONSHIP

AR

The long and short command verbs.

categoryA

A required operand that specifies the category of subject A.

subjectnameA

A required operand that specifies the name of subject A.

WITH

A required operand that enhances the readability of the command.

categoryB

A required operand that specifies the category of subject B.

subjectnameB

A required operand that specifies the name of subject B.

keyword=value

An optional operand that specifies other data about this relationship. See Appendix B for keywords and their values.

Element-Element Relationships

To relate one element to another that is not an alias of the first, the command takes the following form:

```
ADD_RELATIONSHIP ELEMENT subjectnameA +
relationshipkeyword ELEMENT subjectnameB [START=value] +
[BITSTART=n] [USE=value];
```

The parts of the command are described below:

ADD_RELATIONSHIP

AR

The long and short command verbs.

ELEMENT

For element-element relationships, you must use ELEMENT or a valid abbreviation as the category of subject A.

subjectnameA

A required operand that specifies the subject name of subject A. This should be the superior element that contains, redefines, or renames the other.

relationshipkeyword

A required operand that specifies one of these relationships:

CONTAINS

REDEFINES

RENAMES

RENAMES_THRU

RENAMES, REDEFINES, and RENAMES_THRU can also be specified by combinations of CONTAINS, USE, and START.

ELEMENT

For element-element relationships, you must use ELEMENT or a valid abbreviation as the category of subject B.

subjectnameB

A required operand that specifies the name of subject B. This must be the subordinate of two elements in the relationship.

START=value

This keyword must be used whenever the CONTAINS operand is used. It specifies the start position of subject B within subject A. The value is 1 to 5 digits long with a range of 0 to 32767.

BITSTART|BST=n

This keyword is required only if subject B does not start at the beginning of the byte value specified by START. It specifies the starting bit position of subject B in the byte specified by the START value. Valid values are 1 to 8.

USE=value

An optional operand that specifies the meaning of CONTAINS. This operand must be accompanied by START=. The allowable values for this keyword can be any alphabetical character and 6 or 8. The following values give a special meaning to CONTAINS:

C or blank CONTAINS

R REDEFINES

6 RENAMES (with START=1) or RENAMES_THRU (with START=32767)

8 Condition-name (level 88 in COBOL)

For USE=8, the START value stored is zero, no matter what START value is entered.

RENAMES has the same meaning as CONTAINS with USE=6 and START=1. RENAMES_THRU has the same meaning as CONTAINS, USE=6, and START=32767. REDEFINES has the same meaning as CONTAINS with USE=R. These relationship keywords are used to enter COBOL attributes.

Segment-Element Relationships

For segment-element relationships, the general form of the ADD_RELATIONSHIP command is:

```
ADD_RELATIONSHIP SEGMENT subjectnameA {WITH|WITH66} +  
ELEMENT subjectnameB [START=value] [BITSTART=n] +  
[keyword=value] [...];
```

The parts of the command are described below:

ADD_RELATIONSHIP

AR

The long and short command verbs.

SEGMENT

For segment-element relationships, use SEGMENT, or a valid abbreviation, as the category of subject A.

subjectnameA

A required operand that specifies the subject name of subject A.

WITH|WITH66

WITH or WITH66 are required operands that specify the relationship between the two subjects. WITH66 is required if the subjects are COBOL level 66 entries.

ELEMENT

For segment-element relationships, use ELEMENT, or a valid abbreviation, as the category of subject B.

subjectnameB

A required operand that specifies the subject name of subject B.

START=value

A required operand that specifies the start position of the element within this segment. The value is 1 through 5 digits long with a range of 0 to 32767. The value must be 32767 for a COBOL level 66 entry with the WITH66 relationship keyword.

BITSTART|BST=n

This keyword is required only if subject A does not start at the beginning of the byte value specified by START. It specifies the starting bit position of subject B in the byte specified by the START value. Valid values are 1 to 8.

keyword=value

An optional operand that specifies other data about this relationship. See Appendix B for keywords and their values.

To simplify this explanation, subject A is assumed to be the data base or PCB and subject B the segment.. In fact, the order of subject A and subject B is not important.

COBOL Level 66 entries: To establish the proper relationship between a segment and an element that is the subject of a COBOL level 66 item, one of two methods may be used.

- Use the WITH66 keyword to relate the segment and element. This is the preferred method. No other keywords are valid in this form of the command. The appropriate values for the GENFLD and START keywords are implied by WITH66.
- Use the WITH relationship keyword, with which you must specify the keyword GENFLD=6 and START=32767.

Data Base-Segment or PCB-Segment Relationships

To relate a segment to a data base or a PCB, use this form of the ADD_RELATIONSHIP command:

```
ADD_RELATIONSHIP {DATABASE|PCB} subjectnameA +  
WITH SEGMENT subjectnameB [[BEFORE] SEGMENT +  
subjectnameC] [keyword=value] [...] +  
[id=value] [...];
```

The parts of the command are explained below:

ADD_RELATIONSHIP

AR

The long and short command verbs.

DATABASE|PCB

For DATABASE-SEGMENT or PCB-SEGMENT relationships, use DATABASE or PCB or a valid abbreviation as the category of subject A.

subjectnameA

A required operand that specifies the subject name of subject A.

WITH

A required operand that enhances the readability of the command.

SEGMENT

For Data Base-Segment or PCB-Segment relationships, use SEGMENT or a valid abbreviation as the category of subject B.

subjectnameB

A required operand that specifies the subject name of subject B.

BEFORE SEGMENT subjectnameC

An optional operand that specifies the subject name of the segment before which subject B is to be placed. If the operand is omitted, the segment is added at the end of the hierarchy. Mistakes in the hierarchical order may not be noticed until you use the DBD_OUT command. If you find mistakes, you can use the RELOCATE command to change the position of a segment in the hierarchy of a data base or PCB. You can use any valid abbreviation for the word SEGMENT in this operand.

keyword=value

One or more optional keywords that specify more about this relationship. See Appendix B for keywords. Multiple keywords must be separated by blanks.

id=value

A required operand with some keywords that specifies the id keyword and value. See Appendix B.

To simplify this explanation, subject A is assumed to be the data base or PCB and subject B the segment. In fact, the order of subject A and subject B is not important.

Installation-defined Relationships

Use this form of the command to add relationships involving subjects in installation-defined categories. Refer to the Guide reports to determine what relationships may be defined.

```
ADD_RELATIONSHIP categoryA subjectnameA +
relationshipkeyword categoryB subjectnameB +
[sequenceattribute=value] [keyword=value] [...];
```

The parts of the command are described below:

ADD_RELATIONSHIP

AR

The long and short command verbs.

categoryA

A required operand that specifies the category of subject A.

subjectnameA

A required operand that specifies the subjectname of subject A.

relationshipkeyword

A required operand. Use a forward or inverse relationship keyword. Refer to the Guide report appropriate to the installation-defined category you have specified to find the proper relationship keyword. See Chapter 12 of the DB/DC Data Dictionary Applications Guide for examples of how to use the relationship keywords.

categoryB

A required operand that specifies the category of subject B.

subjectnameB

A required operand that specifies the name of subject B.

sequenceattribute=value

An optional operand that identifies the occurrence of a relationship that can occur multiple times between two subjects. This operand is only used when SEQOPT=Y for this relationship-type in the Guide report. The format and valid values for this keyword can be found in the Guide report. See the section titled "Keywords in Commands" at the beginning of this chapter for more information about this keyword.

keyword=value

An optional operand that specifies the subject data to be added. Multiple keywords must be separated by blanks. Keywords are listed in Appendix A.

Adding Aliases

To establish as an alias for a subject, the command takes the following form:

```
ADD_RELATIONSHIP category primaryname {WITH|HAS_ALIAS} +
category alias;
```

The parts of the command are described below:

ADD_RELATIONSHIP
AR

The long and short command verbs.

category

A required operand that specifies the category of the subject. The category can be any except SYS, JOB, PROGRAM, MODULE, PSB, TRANSACTION, SYSDEF, or DDUSER.

primaryname

A required operand that specifies the primary name of the subject.

WITH|HAS_ALIAS

A required operand that specifies the relationship. WITH may be used when the category is DBS, SEG, DTE, or PCB. HAS_ALIAS may be used for any valid category except, SYS, JOB, PROGRAM, MODULE, PSB, TRANSACTION, SYSDEF, or DDUSER.

category

The same category that was specified earlier.

alias

The alias you want to give to the subject name.

CHANGE IN COMMAND

This command changes the subject data associated with a given subject in the Dictionary. Each command specifies the subject with which the data is associated and includes a FROM and a TO operand for the old and new keyword values. The keywords that can be changed by this command are the same subject data keywords that can be added by the ADD command. Multiple keywords may be entered, separated by blanks. The general form of the command is:

```
CHANGE_IN category subjectname [FROM] +
[[keyword=valueA] [...] [id=value] [...]]|EXISTING] +
[TO [[keyword=valueB] [...] [id=value] [...]]];
```

The parts of the command are described below:

CHANGE_IN
CI

The long and short command verbs.

category

A required operand that specifies the category in which information is to be changed.

subjectname

A required operand that specifies the subject name associated with the data to be changed.

FROM
An optional operand.

keyword=valueA
An optional operand identifying a keyword and a value to be changed. See the "Using FROM and TO" below for the rules for modifying keyword values. See Appendix A for keywords and values that may be specified.

id=value
A required operand for some keywords. See Appendix A.

EXISTING
An optional operand specifying that any present value should be changed as indicated by TO keywords. This operand is used for clarity only. It applies to any number of TO keywords.

TO
A optional operand.

keyword=valueB
An operand that specifies the new value of a changed keyword. It is required to change a keyword to a new value, whether the keyword appears in the FROM operand or not. See "Using FROM and TO" below for the rules for modifying keyword values.

id=value
A required operand with some keywords. See Appendix A to determine which keywords require id's.

Using FROM and TO

The FROM and TO operands provide a framework for changing the values of keywords. The FROM operand implies that the values of the keywords associated with it are to be changed. The values for the keywords associated with the TO operand are the values those keywords are to receive. If the keyword whose value you are changing requires an id, you must include the id keyword and value with the FROM operand, the TO operand, or both. You may omit the FROM or the TO operand and specify only keywords and ids and their values, but the changes you may make without using the operands are limited. The table below list all possible uses of the FROM and TO operands and the keywords and ids associated with them. Brackets indicate optional operands that do not affect the results of the command, but that may be included to enhance readability.

From Expression	To Expression	Action
[FROM]keyword= value A	TO keyword=value B	Change to value B of current value is value A.
[FROM]keyword=value A	[TO]	If current value A, make alpha- meric fields blank, make num- eric fields zero.
[FROM][EXISTING]	TO keyword=value B	Change to value B regardless of current value.
[FROM][EXISTING]	[TO]	No change.

Note: If you do not specify FROM, specify TO with a keyword and value, and the keyword you specify has not yet been added, there is a possibility of two different results when you execute the command. If the category you specified is installation-defined or if segment in the Dictionary data base containing this

keyword has already been added for this subject, the keyword value will be added. If the category you specified is standard and the segment was not added, you will receive a warning message.

CHANGE NAME COMMAND

This command changes an existing subject name to a new subject name. In addition, the Dictionary optionally searches other entries that refer to the subject name and changes the references.

The command can be used to change either a primary name or a secondary name (alias).

The CN command cannot be used to change the status of a subject. The COPY command with the AFFECTED option should be used for this purpose.

The general form of the command is:

```
CHANGE_NAME category subjectnameA [TO] category +
subjectnameB [ISR={YES|NO}];
```

The parts of the command are described below:

CHANGE_NAME
CN

The long and short command verbs.

category

A required operand that specifies the category of the old subject name.

subjectnameA

A required operand that specifies the old subject name.

TO

An optional operand.

category

The same category specified earlier.

subjectnameB

A required operand that specifies the new subject name. The new name must not duplicate any existing name in the specified category.

ISR=YES|NO

An optional operand that specifies whether the Dictionary is to be scanned for indirect subject references. If NO is specified, subjectnameA remains unchanged wherever it appears as an indirect subject. If the operand is omitted, the default is YES. See the section on ISR reports in the "REPORT Command" section later in this chapter for the names of the fields that are searched for indirect references.

CHANGE_RELATIONSHIP_DATA COMMAND

This command changes relationship data associated with a given pair of related subjects. Each command specifies the two related subjects and their categories and may include a FROM and a TO operand for the old and new keyword values respectively. The keywords changed by this command are the same relationship data keywords that can be added by the ADD_RELATIONSHIP command. Multiple keywords may be entered, separated by blanks.

The general form of the command is:

```
CHANGE_RELATIONSHIP_DATA categoryA subjectnameA +
[WITH|relationshipkeyword] categoryB subjectnameB +
[FROM] [[keyword=value] [...] [id=value] [...]] +
[sequenceattribute=value]]|EXISTING]
[TO [[keyword=value] [...] [id=value] [...]] +
[sequenceattribute=value]]];
```

The parts of the command are described below:

CHANGE_RELATIONSHIP_DATA **CRD**

The long and short command verbs.

categoryA

A required operand that specifies the category of subject A.

subjectnameA

A required operand that specifies the subject name of the first related subject.

WITH|relationshipkeyword

For relationships between standard categories you must specify the WITH operand.

For relationships in which either of the subjects are in installation-defined categories you must use a forward or inverse relationship keyword. Refer to the Guide report appropriate to the installation-defined category you have specified to find the proper relationship keyword. See Chapter 12 of the DB/DC Data Dictionary Applications Guide for examples of how to use the relationship keywords.

categoryB

A required operand that specifies the category of subject B.

subjectnameB

A required operand that specifies the subject name of the second related subject.

FROM

An optional operand.

keyword=value

A required operand identifying a keyword and a value to be changed. See the "Using FROM and TO" in the CHANGE_IN command for the rules for modifying keyword values. See Appendix B for keywords and values that may be specified.

id=value

A required operand for some keywords. See Appendix B.

sequenceattribute=value

A required operand for installation-defined relationships which are sequenced. See the Guide report for the name of the sequence attribute and its valid values. See the

"Keywords in Commands" section at the beginning of this chapter for more information about the sequence attribute keyword.

EXISTING

An optional operand specifying that any value present should be changed as indicated by TO keywords. This operand is used for clarity only. It applies to any number of TO keywords.

TO

A optional operand.

keyword=value

An operand that specifies the new value of a changed keyword. It is required to change a keyword to a new value, whether the keyword appears in the FROM operand or not. See "Using FROM and TO" in the CHANGE_IN command for the rules for modifying keyword values.

id=value

A required operand with some keywords. You must specify a value for the id START (and BITSTART if the START position is not on a byte boundary) if either of the subject names in the command has a category of ELEMENT. See Appendix B.

sequenceattribute=value

A required operand for installation-defined relationships which are sequenced.

CHECKPOINT COMMAND

The Checkpoint command is used to request that a DL/I checkpoint be taken. It may be used in batch mode or online under CICS/VS.

The general format of the command is:

```
CKPT;
```

CKPT

The command verb.

COBOL IN COMMAND

The COBOL_IN command looks at COBOL source statements and extracts descriptions of records and segments and the groups and elements within them. It produces Dictionary entries for subjects and the structure of the records. It may only be used in batch mode.

The general form of the command is:

```
COBOL_IN [MEMBER={name}(name1,name2,...name10)] +  
[LIST={NO|SOURCE|ENTRIES|ALL}] [SEGLVL={1|n}] +  
[UPDATE={YES|NO}] [LANGUAGE={A|B|C|J|K|L}] +  
[DEST={P|L}];
```

The parts of the command are described below:

COBOL_IN
CBI

The long and short command verbs.

MEMBER|MEM|BOOK=

An optional operand that specifies that input is from an OS/VS partitioned data set or DOS/VS source statement library instead of a sequential file. If this operand is omitted, input from a sequential data set is assumed. The default is the CBISEQ data set for IMS/VS. The default is the sequential file in SYS004 for DOS/VS.

name|(name1,name2,...name10)

This operand specifies the names of up to ten members of a partitioned data set or a source statement library. Each name must be a valid member or book name. The data set is defined by the CBIPDS DD statement for IMS/VS and by sublibrary C of SYSSLB for DL/I DOS/VS.

LIST=N|NO|S|SOURCE|E|ENTRIES|A|AL|ALL

An optional operand that specifies the system output desired. Choose one or omit. If this operand is omitted, NO is assumed. The valid values are:

- NO** The abbreviation is N. Only errors, subjects not entered under default occurrence number, and unique FILLER names are listed. This value is the default.
- SOURCE** This operand requests a listing of the COBOL source statements. It also produces the output specified by the NO operand. The abbreviation is S.
- ENTRIES** This operand requests a listing of all commands that were generated to add the structure to the Dictionary. The list that is generated by the NO option is not produced. The abbreviation is E.
- ALL** This operand specifies that both SOURCE and ENTRIES reports are desired. The abbreviation is A or AL.

SEGLVL|SL=1|n

An optional operand that specifies the COBOL level number to be used by COBOL_IN when processing of the input structures. Data items in the structure having a level number less than or equal to n will be entered into the Dictionary as SEGMENTS. The default segment level is 1.

UPDATE=Y|YES|N|NO

This operand specifies whether the structure processed by the command is to be actually entered into the Dictionary. YES or Y specifies that the segment and element definitions and interrelationships corresponding to the structure are to be entered into the Dictionary. NO or N specifies that the structure be evaluated against the current contents of the Dictionary, but that the Dictionary not actually be updated. If omitted, the default is NO.

LANGUAGE|LANG=A|B|C|J|K|L

An optional operand that specifies the language code to be assigned to each segment and element entered into the Dictionary. The valid values are:

- A - Assembler (Used in DL/I)
- B - PL/I (Used in DL/I)
- C - COBOL (Used in DL/I)
- J - Assembler (Non-DL/I)

K - PL/I (Non-DL/I)

L - COBOL (Non-DL/I)

LANG=C is the default for this command regardless of what the current default value is for language code. Specifying A is to avoid the need for building DL/I alias names when the definitions from COBOL_IN are to be integrated with PSB and DBD source statements. If you specify A, all names must be no more than eight characters long. The structures can be obtained in COBOL format with the STRUCTURES_OUT command by specifying a COBOL output language option.

DEST=P|L

This operand specifies the destination of the command output that is generated when UPDATE=NO and LIST=ENTRIES or ALL are specified. The valid values are:

P Punch

L Line printer

If this field is omitted, the default is L.

Refer to Chapter 9 of the DB/DC Data Dictionary Applications Guide for a list of restrictions on the COBOL_IN command.

COPY COMMAND

This command copies all the subject data and aliases stored in the Dictionary for a specified subject name and stores the data under a new subject name. In addition, you can request that the copy include relationship entries. The COPY command can also copy subjects in a Dictionary hierarchical structure, proceeding upward or downward from a specified subject. For a detailed discussion and examples of the COPY command refer to Chapter 5 of the DB/DC Data Dictionary Applications Guide.

The general form of the command is:

**COPY categoryA subjectnameA [TO] categoryB subjectnameB +
[AND][RELATE|DEPENDENTS|AFFECTED];**

COPY

The command verb.

categoryA

A required operand that specifies the category of the subject to be copied.

subjectnameA

A required operand that specifies the subject name of the subject to be copied.

TO

An optional operand that is used for clarity only.

categoryB

A required operand that specifies the same category that was used for categoryA.

subjectnameB

A required operand that specifies the subject name under which the copy is to be stored.

AND

An optional operand.

RELATE|DEPENDENTS|AFFECTED

These optional keywords specify the relationships and related subjects to be included with the copy. They cannot be used with relationships involving installation-defined categories.

RELATE Copies all subject data and aliases as well as all downward relationships (but no subjects). The downward relationships are copied according to the status rules listed below.

DEPENDENTS Copies all subject data and aliases. Additionally, all the subjects and relationships that exist downward in the Dictionary hierarchical structure are copied, constrained only by the naming and status rules listed below.

AFFECTED Copies all subject data and aliases as well as all downward relationships (but no subjects). Also each hierarchically superior subject and its downward relationships are copied, constrained only by the naming and status rules given below.

If no option is specified, the command creates a copy of subjectnameA with the name subjectnameB with all subject data (attributes, description, user text, etc.) included, but no relationships are copied.

Aliases are also copied if a unique alias name can be created by substituting the subjectnameB status code for the existing alias status code.

If there are no errors in the COPY command, subjectnameA is always copied and given the name subjectnameB. If subjectnameB duplicates the name of an existing subject, see the third naming rule below.

NAMING AND STATUS RULES FOR COPYING SUBJECTS: When a structure is being copied, each subject in that structure is a candidate for copying; a subject name in the structure is called a candidate name. The Dictionary builds a trial name for the copy by replacing the candidate name's status code with the status code of the new name (specified in the COPY command). No other change is made to the candidate name to create a trial name. If the trial name is:

- The same as the candidate name, the candidate subject is not copied. No other subjects in the hierarchical path are copied.
- Different from the candidate name and does not name an existing subject, the candidate subject is copied and the copy is given the trial name.
- Different from the candidate name and names a subject that already exists, the existing subject, its data, and its downward relationships are deleted. The candidate subject is copied and the copy is given the trial name. All upward and nonstructural relationships participated in by the deleted subject are retained.

As the new structure is being built, the Dictionary notes the status code of each new subject. A relationship between a subject in the new structure and one in the original structure is established only when it does not violate these rules:

- Within a Dictionary structure, no more than two status codes can exist.
- If two status codes exist in a structure, one of them must be P (production).
- Subjects with P (production) status can be subordinate to other P status subjects or to test status subjects.
- Subjects with a test status can never be subordinate to subjects with P status.

Note: The Dictionary allows you to create relationships that are never copied by the COPY function. Such relationships, called nonstructural relationships, might be useful for describing a data processing resource. However, they relate two subjects (for example, system-to-SYSDEF.) to which the COPY function attaches no hierarchical significance. Nonstructural relationships are not copied and are not followed to obtain copies of related subjects. When such relationships are encountered by the COPY function, they are reported with a message and are then skipped.

Relationships that are nonstructural are:

- Relationships between subjects within the same category of the system data base.
- Relationships between subjects in selected categories in the system data base; that is, SYSDEF with SYSTEM, JOB, PROGRAM, or MODULE, and TRANSACTION with MODULE.
- Relationships of an installation-defined relationship type.

DBD_IN COMMAND

This command is used to retrieve DBDs and read them into the Dictionary. A command is necessary for each DBD, but any number of DBD_IN commands may be processed in a single computer run.

DBD_IN is a batch-only command.

The general form of the command is:

```
DBD_IN subjectname MODE=n [GEN={ALL|SEQ}] +  
[RELATE={YES|NO}];
```

DBD_IN
DI

The long and short command verbs.

subjectname

A required operand that specifies the subject name to be stored. The subject code (data base type) can be:

P	Physical
L	Logical
F	Non-DL/I data set
X	Primary index
Y	Secondary index

The occurrence number is always zero for DL/I data bases. The user name is the 1-8 alphanumeric character name of the DBD in the DBD library.

MODE=n

A required operand that specifies the action to be taken if a subject name already exists in the Dictionary. The valid values are:

1	Create a new entry
2	Overlay existing entry
3	Do not create a new entry, use the existing entry

For more information about the use of the mode operand, see Chapter 6 of the DB/DC Data Dictionary Applications Guide.

GEN=ALL|SEQ

This operand specifies which fields are to be marked for processing by the DBD_OUT command (with nonblank characters in GENFLD keyword value, see Appendix B). Allowed values are:

ALL	All fields
SEQ	Only sequence fields

If the operand is omitted, the default is SEQ.

RELATE=YES|NO

An optional operand that specifies whether DBS-DTE relationships should be created. YES indicates the relationships should be created; NO indicates the relationships should not be created. If this operand is omitted, the default is YES.

See Chapter 6 of the DB/DC Data Dictionary Applications Guide for restrictions on the use of the DBD_IN command.

DBD_OUT COMMAND

This command is used to search the Dictionary and prepare source statements in card-image form to update DL/I DBD libraries. It can also produce line printer output or display terminal output.

The general form of the command is:

```
DBD_OUT subjectname [GEN={ALL|MARKED}] +
[DLILVL=value] [GSAMDBD] [DEST={P|L|T}];
```

DBD_OUT
DO

The long and short command verbs.

subjectname

A required operand that specifies the subject name of the data base or data set. Allowed values for the subject code part of the name are:

P	Physical
L	Logical
F	Non-DL/I data set
X	Primary index
Y	Secondary index

The occurrence number is always zero for data bases. The user name is the 1-8 alphameric character name of the output DBD. It may be the primary name of the data base or an alias.

GEN=ALL|MARKED

An optional operand that specifies the elements that are to be included in the output DBD. The valid values are:

ALL	All elements related to the segments in this data base
MARKED	Only those elements marked for DBD_OUT

If the operand is omitted, the default is MARKED.

DLI LVL=value

An optional operand that specifies the DL/I version for of the DBDs to be produced. Dictionary attributes for a version later than specified will be ignored. Allowed values for this operand are:

1.0	IMS/VS Version 1.0
1.1	IMS/VS Version 1.1 and later
D1.1	DL/I DOS/VS Version 1.1 or 1.2
D1.3	DL/I DOS/VS Version 1.3 or later

If the value is omitted, the default is 1.1 or D1.3.

GSAMDBD

An optional operand that specifies a GSAM DBDGEN. It is used only for physical data bases or non-DL/I OS files. It is not necessary for GSAM if data base definition includes ACCESS=GSAM.

DEST=P|L|T

An optional operand that specifies the output device. Valid values are:

P	Punch
L	Line printer
T	Terminal

If the value is omitted, the default is P.

When GSAMDBD is specified, this keyword overrides the access parameter of the data base or file specified by the subject name, and the command produces a GSAM DBDGEN. GSAMDBD thus can be used to produce a GSAM

DBDGEN. GSAMDBD thus can be used to produce a GSAM DBDGEN for any physical data base or OS file defined in the Dictionary. The GSAMDBD will have the following attributes of the OS file or data base:

INDACCS (optional)	BLKSIZE1 (optional)
PASSWD (optional)	RCDSIZE1 (optional)
DD1	RCDSIZE2 (optional)
DD2 (optional)	RECFM

When GSAMDBD is specified, the DLILVL keyword and related segments and fields are ignored. If GSAMDBD is specified for a logical data base, the keyword is ignored and a logical DBDGEN results. If the data base definition in the Dictionary includes the ACCESS=GSAM parameter, GSAMDBD should be omitted from the command.

When GSAMDBD is specified, the ACCESS parameters of the file or data base may be used to determine the secondary ACCESS parameter of the GSAM DBD, as shown in Figure 65.

Value of DBSACCS	Value of INDACCS	Resulting Value for ACCESS
VSAM	Any value	(GSAM,VSAM)
BSAM	Any value	(GSAM,BSAM)
Any other value	VSAM	(GSAM,VSAM)
Any other value	BSAM	(GSAM,BSAM)
Other	Other	GSAM

Figure 66. Effect of GSAMDBD Operand in DBD_OUT Command on ACCESS=Value

DDT_OUT COMMAND

The DDT_OUT command is used to obtain a set of Data Definition Table (DDT) source statements suitable for editing into the source input stream for GIS DDT generation. It causes the set of fixed length records, which contain sequence numbers, to be created from the contents of the Dictionary data base, PCB, segment, and field categories for a specified data base and PCB combination.

The general format of the command is:

```
DDT_OUT dbname [PCB pcbname] [FILENAME=filename] +
[SYSTEM={OS|DOS}] [DEST={P|L|I}];
```

DDT_OUT

The command verb.

dbname

A required operand that signifies the subject name of the data base category for which the GIS DDT is to be produced.

PCB

An optional operand that must be immediately followed by the value for **pcbname**.

pcbname

This operand specifies the fully qualified name of the PCB subject from which the Dictionary is to obtain processing sequence and indexing information. If SYSTEM=OS is specified, the PROCSEQ, PCBPARM, INDICES, and XDFLD parameters are produced. If SYSTEM=DOS is specified, PROCSEQ and XDFLD parameters are produced. Also, the Dictionary will produce SEGM statements based on PCB-SEG relationships. Should the PCB specified have values associated with the query and update security parameters, these values will be produced.

This operand **must** be specified for DL/I data bases with secondary index fields.

If the PCB and pcbname combination is not specified, the Dictionary will only use DBS-SEG relationships to generate the DDT SEGM statements. Any query or update security values associated with the DBS will be used.

FILENAME=filename

This optional operand specifies the special data set or master file name to be used with the NAME keyword in the generated DDT's FILE statement. This file name must begin with an alphabetic character; it can be from 1 to 8 characters in length.

If the FILENAME operand is not specified, the data base name specified for **dbname** will be used with the NAME keyword and any assembler aliases will be used by the Dictionary to generate the SYNM statements for the DDT. If the FILENAME operand is specified, all aliases will be ignored and no SYNM statements will be generated.

SYSTEM=OS|DOS

An optional operand that specifies the operating system for which the DDT is to be generated. Valid values are OS or DOS. If this operand is not specified, the Dictionary will generate a DDT which is compatible with the operating system on which the Dictionary is executing.

DEST=P|L|T

An optional operand that specifies the destination of the command's output. Valid values are:

- P The output from this command is to be routed to the punch.
- L The output from this command is to be routed to the printer.
- T The output from this command is to be routed to the display terminal.

If the value is omitted, the default is P.

For additional information and examples of the DDT_OUT Command see Chapter 11 of the DB/DC Data Dictionary Applications Guide.

DEFAULTS COMMAND

The DEFAULTS command is used to obtain the current status of the Dictionary defaults. The full set of responses to the command is shown in Figure 67.

The format of the command is:

DEFAULTS;

DEFAULTS

The command verb.

The use of the command in batch is solely for information. When entered online from the COMMAND form, the set of response messages can be used to verify defaults before entering SET commands to change default values for subsequent use in commands or on display forms.

The figure below shows all the Dictionary defaults with dummy values. The values that will appear are the values your installation defined when the Dictionary was installed. If you change the default values with the SET commands, the changed values will appear in place of the installation-defined values. See Chapter 1 of the DB/DC Data Dictionary Administration and Customization Guide for more information about Dictionary defaults.

```

DBD5100 I  DICTIONARY DEFAULTS:
DBD5100 I  STATUS . . . . .x
DBD5100 I  LANGUAGE . . . . .x
DBD5100 I  DATABASE CODE. . . . .x
DBD5100 I  OCCURRENCE . . . . .nnn
DBD5100 I  FLUSH. . . . .x
DBD5100 I  LOG USER-ENTERED CMNDS . . . .x
DBD5100 I  LOG INTERNAL COMMANDS. . . .x
DBD5100 I  STR-OUT LEVEL TWO. . . . .nn
DBD5100 I  STR-OUT LEVEL INCREMENTS . . .nn
DBD5100 I  SEGMENT ATTRIBUTES . . . . .xxxxxxxxxxxxxxxxxxx
DBD5100 I  SCAN LIMIT (BATCH) . . . . .nnnnn
DBD5100 I  SCAN DEFAULT (BATCH) . . . . .nnnnn
DBD5100 I  SCAN LIMIT (ON-LINE) . . . . .nnnnn
DBD5100 I  SCAN DEFAULT (ON-LINE) . . . . .nnnnn
DBD5100 I  USER DATA FORM . . . . .nn
DBD5100 I  SECURITY ENABLED . . . . .x
DBD5100 I  IMS/CICS SECURITY ENABLED . . .x
DBD5100 I  SECURITY STATUS CODE . . . . .x
DBD5100 I  SIGN-ON ATTEMPT THRESHOLDtop .nn
DBD5100 I  CKPT FREQUENCY (SECONDS) . . .nnn

```

Figure 67. Response to the DEFAULTS Command

DELETE COMMAND

This command is used to delete a subject name and its definition from a specified category. All the information, except aliases, associated with the subject in the specified category will be deleted, including primary name, attributes, description, User Data, and relationship entries that establish relationships between this subject and other subjects. In addition, relationship data will be deleted. Aliases are left in the Dictionary; since their primary name has been deleted, they will have no data associated with them. The names and definitions of the previously related subjects will be unaffected. If an alias is specified, only the alias is deleted, the alias' primary name is unaffected.

The general form of the command is:

```
DELETE category subjectname [UNRESTR];
```

DELETE DEL

The long and short command verbs.

category

A required operand that specifies the category of the subject to be deleted.

subjectname

A required operand that specifies the subject name of the subject to be deleted. If the subjectname is an alias, only the alias will be deleted.

UNRESTR

An optional operand that specifies that the delete operation is not restricted to subjects without aliases. If this operand is omitted, an attempt to delete a subject with an alias results in an error message.

If the operand is specified, a subject name with aliases will be deleted. (The aliases however, will not be deleted.)

DELETE DATA COMMAND

This command is used to delete subject data for a given subject. With it, you can delete subject attributes, description, and user data for entire segments in standard categories. You can only delete description text and user data for subjects in installation-defined categories.

The general form of the command is:

```
DELETE DATA category subjectname +  
DATA=([segmentname[,id]],...);
```

DELETE_DATA

DD

The long and short command verbs.

category

A required operand that specifies the category from which data is to be deleted.

subjectname

A required operand that specifies the subject name associated with the data to be deleted.

DATA=

A required operand.

segmentname The name of the Dictionary segment to be deleted. The segment names are given for each category at the top of each table in Appendix A. Each table shows the group of data that is contained as subject data in the segment. A maximum of 10 segment names can be specified. For subjects in installation-defined categories, only description (DESC) or User Data (USERn|EXTUSERn, where n=1 to 5) segments may be specified.

id

This operand is the id value necessary to identify some segments. See Appendix A.

If the Dictionary segment has an id, its value must be included in the DATA= operand. Only one segment may be deleted for each segment name specified. You cannot delete all occurrences of DBSALLOC, for example, by coding DBSALLOC without specifying the id value. Each occurrence to be deleted must be explicitly specified, but the same segment name may be repeated, with a different id in a command.

DELETE RELATIONSHIP COMMAND

This command deletes a relationship between two Dictionary subjects. The command is the same for aliases, standard relationships between two subjects, relationship entries that include relationship data or relationships involving a subject in an installation-defined category. The command deletes the relationship entry for both of the subjects involved in the relationship, and all the relationship data associated with the relationship entry, but does not affect the subject data.

The general form of the command is:

```
DELETE_RELATIONSHIP categoryA subjectnameA +
relationshipkeyword categoryB subjectnameB [START=value] +
[BITSTART=n] [sequenceattribute=value];
```

DELETE_RELATIONSHIP

DR

The long and short command verbs.

categoryA

A required operand that specifies the category of subject A.

subjectnameA

A required operand that specifies the name of subject A. If you are deleting an alias, this is the primary name.

relationshipkeyword

A required operand that specifies the relationship keyword. Valid values for relationships between standard categories are:

WITH
HAS_ALIAS
WITH66
RENAMES
RENAMES_THRU
CONTAINS
REDEFINES

If one or both of the subjects are in installation-defined categories you must use a forward or inverse relationship keyword. See Chapter 12 of the DB/DC Data Dictionary Applications Guide for examples of how to use the relationship keywords.

categoryB

A required operand that specifies the category of subject B. If you are deleting an alias, use the subject A category.

subjectnameB

This operand specifies the subject name of subject B. If you are deleting an alias, specify the alias here.

START=value

Specifies the start position of subject B within subject A. Used for ELEMENT/ELEMENT and SEGMENT/ELEMENT relationships. Valid values are 1 to 5 digits long with a range from 0 to 32767.

BITSTART|BST=n

This keyword is required only if subjectnameB does not start at the beginning of the byte value specified by START. It specifies the starting bit position of subjectB in the byte specified by the START value. Valid values are 1 to 8.

sequenceattribute=value

A required operand for installation-defined relationship-types which are sequenced.

If the relationship you are deleting is an alias relationship, you must specify the primary name as subject A.

If the relationship you are deleting is a CONTAINS, RENAMES, RENAMES_THRU, or REDEFINES relationship between elements, subject A must be the superior (the containing, renaming, or redefining) element.

If the relationship is segment WITH element or element CONTAINS element, the id keyword START must be specified.

DELETE RELATIONSHIP DATA COMMAND

This command is used to delete relationship data (for data base-segment and PCB-segment relationships) from the Dictionary data bases. The information to be deleted is specified by Dictionary segment name (the name of the segment that stores this relationship data in the Dictionary data base).

The general form of the command is:

```
DELETE_RELATIONSHIP_DATA DATABASE|PCB subjectnameA +  
WITH SEGMENT subjectnameB +  
DATA=((segmentname[,id]),...);
```

DELETE_RELATIONSHIP_DATA **DRD**

The long and short command verbs.

DATA BASE|PCB

A required operand that specifies the category data base or PCB. Specify DATABASE, PCB, or a valid abbreviation.

subjectnameA

A required operand that specifies the subject name of a data base or PCB.

WITH

An required operand.

SEGMENT

A required operand. Use SEGMENT or any valid abbreviation.

subjectnameB

A required operand that specifies the subject name of the segment to which the data base or PCB is related.

DATA=

A required operand.

segmentname A required operand that specifies the name of the segment to be deleted. See Appendix B for the segment names. They are given in parentheses at the top of each table.

id A required operand for some segments that specifies the value of the id keyword necessary to identify some segments. See Appendix B for requirements.

If the Dictionary segment has an id keyword, its value must be included in the DATA= operand; if it has no id keyword, the id value specification is omitted. For Dictionary segments that require an id keyword, only one id value at a time can be deleted. You cannot delete all occurrences of SNSGINDC, for example, by coding SNSGINDC without specifying the id value. Each occurrence to be deleted must be explicitly specified (one id per segment name, but the same segment name may be repeated more than once in

a command). Up to ten segment names, each with one id, can be specified in one command.

To simplify this explanation, subject A is assumed to be the data base or PCB and subject B the segment. In fact, the order of subject A and subject B is not important.

DELETE STRUCTURE COMMAND

This command deletes part of a structure or an entire structure from the Dictionary, beginning with the subject specified and continuing down the structure to the subordinate level specified. All associated aliases are also deleted. Chapter 5 of the DB/DC Data Dictionary Applications Guide contains a diagram of the Dictionary structure that the DELETE_STRUCTURE command operates on.

The general form of the command is:

```
DELETE_STRUCTURE categoryA subjectname [STOP=categoryB];
```

DELETE_STRUCTURE

The command verb.

categoryA

A required operand that specifies the category of the subject with which the deletion is to begin. The category specified may not be installation-defined.

subjectname

A required operand that specifies the subject name with which the deletion is to begin. This subject will always be deleted.

STOP=categoryB

An optional operand that specifies the level at which to stop deleting. Subjects in the specified category are not deleted. If this operand is omitted, the structure is deleted through the lowest related level (subelement, if any). The command will not delete the STOP level specified or any lower levels in the hierarchy. For example, you might enter a DELETE_STRUCTURE command specifying deletion of a program, with a STOP specified at the data base level. The command would delete the program and all PSBs, modules, transactions, and PCBs in the hierarchic path, but no data bases, segments or elements.

The DELETE_STRUCTURE command works with Dictionary structures rather than with single subjects. For example, one command can delete a data base, all related segments, and all related elements. The command deletes the specified subject and all subordinate subjects if they are not related to higher-level subjects that are not being deleted. In a deletion beginning with a data base, a subordinate segment will not be deleted if it is related to other data bases. Only the segment's relationship with the specified data base (and any associated relationship data) will be deleted. Special cases are: 1) the PSB, which will not be deleted if it is related to more than one subject in the System data base (other than

another PSB), 2) the data base, which will not be deleted if it is related to more than one PCB, SYSDEF, or PGM and 3) the transaction, which will not be deleted if it is related to more than one SYSDEF or PGM.

If a subject cannot be deleted because it is related to other superior subjects, its dependents will not be deleted either. Subject names with production status will be deleted only if the subject name in the command also has production status.

Note: This command can delete large amounts of Dictionary data. You should understand the hierarchy of Dictionary subjects before using this command. See Chapter 5 of the DB/DC Data Dictionary Applications Guide for more information about Dictionary hierarchy and the DELETE_STRUCTURE command.

EXECUTE COMMAND

The EXECUTE command is used to execute a user program that may retrieve data from Dictionary data bases, validate data base contents, create user-defined reports, punch out Dictionary commands, or make some other use of Dictionary data.

The format of the command is:

```
EXECUTE PGM=pgmname [PARAM='parmstring'];
```

EXECUTE EXEC

The long and short command verbs.

PGM=pgmname

This is a required operand that specifies the name (pgmname) of the user routine. The name can be 1 to 8 characters long but must begin with an alphabetic character. Consult with the data base administrator or system programmer at your installation for information about the programs available.

PARAM='parmstring'

An optional operand that is used to pass a string of parameters to the program. The maximum length of the parameter string is 200 characters. Consult with your data base administrator or system programmer for information about valid parmstrings for programs at your installation.

EXTEND RELATIONSHIP COMMAND

The EXTEND_RELATIONSHIP command relates all the fields of a segment to a given subject (such as a data base) or relates all the first level subfields of a field to a given subject (such as a segment). Additionally, if the relationship between the segment and the other subject (or the field and the other subject) does not exist, it will be established. The segment (or field) whose fields (or subfields) are being related to the other subject is called the source subject. The other subject is called the target subject.

The general form of the command is:

```
EXTEND_RELATIONSHIP category subjectnameA [TO] +
targettype subjectnameB + [[BEFORE] SEGMENT segmentnameC] +
[RELATE={SEQ|GEN}] [START=value][BITSTART=n];
```

EXTEND_RELATIONSHIP

XR

The long and short command verbs.

category

This required operand must be SEGMENT or FIELD or any valid abbreviation.

subjectnameA

This operand specifies the subject name of the segment or field whose fields are to be related to the target subject, subjectnameB.

TO

This is an optional relation keyword. If it is omitted, the meaning of the command is the same.

targettype

The operand specifies the subject type of the target subject. The categories listed below and their abbreviations are valid values for this field:

SEGMENT	segment
DATABASE	data base
PCB	Program Control Block
PSB	Program Specification Block
SYSTEM	system
JOB	job
PROGRAM	program
MODULE	module
TRANSACTION	transaction
SYSDEF	system definition

subjectnameB

This operand specifies the name of the Dictionary subject to which subjectnameA and related fields are to be related.

BEFORE SEGMENT segmentnameC

These optional keywords are used only if targettype is DATABASE or PCB and the source is SEGMENT. If this keyword is not specified, the relationship between the segment and data base or PCB will be established at the end of the other segments already related to the data base or PCB. If this keyword is specified, the relationship will be established before the segment specified by the segmentname field.

RELATE=SEQ|GEN

This optional keyword specifies what fields in the source segment are to be related to the target subject. Use it only when the source is SEGMENT. SEQ relates only sequence fields to the

target subject. GEN relates all fields that are included in the DBD to the target subject. If this keyword is omitted, GEN is the default.

START=value

If the target is a SEGMENT, this operand specifies the starting position in the target subject that the source subject is to assume. Only specify this operand if the target subject is a SEGMENT.

BITSTART|BST=n

This keyword is required only if subject B does not start at the beginning of the byte value specified by START. It specifies the starting bit position of subject B in the byte specified by the START value. Valid values are 1 to 8.

If the source subject is a SEGMENT and the target subject is another SEGMENT, the source is not related to the target because it is invalid to relate a SEGMENT to a SEGMENT except by an alias relationship.

If the source subject is a FIELD, it relates all of the first-level subfields related to the target subject. The target subject cannot be another FIELD.

THE FLUSH COMMAND

This command lets you specify the action you want the system to take when it detects an error (severity E or S) in a command. The command may be used at any point in a job stream, and as many times as is necessary. The command is:

FLUSH={Y|YES|N|NO}

where YES or Y means the system should, on detecting an error, "flush" all subsequent commands; that is, process them for syntax errors only, but execute none but the REPORT command.

FLUSH=NO means each command following an error should be executed. If you do not specify FLUSH, the default is whatever your installation has defined.

If an error is found while processing batch form records and FLUSH=YES has been specified, subsequent detail records will be checked for syntax errors but not processed. However, if an error is found while processing a header record, only syntax will be checked, regardless of the FLUSH command setting.

The FLUSH command is used for batch operation only (including IMS/VS batch message processing), since in online operation each command is processed as it is entered and you have immediate feedback before entering another command.

If FLUSH appears more than once on the same line, it is processed separately each time. You may wish to use the command sequence FLUSH=NO followed by FLUSH=YES to stop flushing at a certain point (after a group of related commands) and resume processing of commands unless another error appears.

If the error has severity T (terminating), the command will be flushed without syntax checking, regardless of the current FLUSH status.

PLI IN COMMAND

The PLI_IN command accepts only those PL/I source statements which describe structures. The source statements become Dictionary entries in the segment and element categories, and they also become Dictionary structures. The Dictionary structure is represented as relationships between elements and relationships between segments and elements.

PLI_IN is a batch-only command.

The general form of the command is:

```
PLI_IN [MEMBER={name|(name1,name2,...name(10))}] +  
[LIST={NO|SOURCE|ENTRIES|ALL}] [SEGLVL={1|n}] +  
[UPDATE={YES|NO}] [LANGUAGE={A|B|C|J|K|L}] +  
[DEST={P|L}];
```

PLI_IN

The command verb. There is no abbreviated form.

MEMBER|MEM|BOOK=

An optional operand that specifies that input is from an OS/VS partitioned data set or from a DOS/VS source statement library. The operand can be abbreviated to MEM. If this operand is omitted, input from a sequential data set is assumed. The default data set name is PLISEQ for OS/VS. The default is the sequential file in SYS005 for DOS/VS.

name|(name1, name2,...name10)

This operand specifies the names of up to ten members of the OS/VS partitioned data set or DOS/VS source statement library. The name is 1-8 alphanumeric characters. The data set is defined by the PLIPDS DD statement for IMS/VS, and by sublibrary P of SYSSLB for DOS/VS.

LIST=N|NO|S|SOURCE|E|ENTRIES|A|AL|ALL

An optional operand that specifies the Dictionary output desired. Choose one or omit. If this operand is omitted, NO is assumed. The valid values are:

- | | |
|----------------|---|
| NO | The abbreviation is N. Only errors, subjects not entered under default occurrence number, and unique FILLER names are listed. This value is the default. |
| SOURCE | This operand requests a listing of the PL/I source statements. It also produces the same output that NO does. The abbreviation is S. |
| ENTRIES | This operand requests a listing of all commands that were generated to add the structure to the Dictionary. The output generated by the NO option is not produced. The abbreviation is E. |
| ALL | This operand specifies that both SOURCE and ENTRIES reports are desired. The abbreviation is A or |

AL.

SEGLVL|SL=1|n

An optional operand that specifies the PL/I structure level number to be used by PLI_IN when processing the input structures. Data items in the structure having a level number less than or equal to n will be entered into the Dictionary as SEGMENTS. The default segment level is 1.

UPDATE=Y|YES|N|NO

This operand specifies whether the structure processed by the command is to be entered into the Dictionary. YES or Y specifies that the segment and element definitions and interrelationships corresponding to the structure are to be entered in the Dictionary. NO or N specifies that the structure is to be evaluated against the current contents of the Dictionary, but that the Dictionary is not actually to be updated. If omitted, the default is NO.

LANGUAGE|LANG=A|B|C|J|K|L

An optional operand that specifies the language code to be assigned to each segment and element entered into the Dictionary. The valid values are:

- A - Assembler (Used in DL/I)
- B - PL/I (Used in DL/I)
- C - COBOL (Used in DL/I)
- J - Assembler (Non-DL/I)
- K - PL/I (Non-DL/I)
- L - COBOL (Non-DL/I)

LANG=B is the default if this parameter is omitted. The effect of specifying A is to avoid the necessity for building DL/I alias names when the PL/I definitions are to be integrated with PSB and DBD source statements. If you specify A, all names must be no more than eight characters long. The structures can be obtained in PL/I format with the STRUCTURES_OUT command by specifying a PL/I output language option.

DEST=P|L

This operand specifies the destination of the command output that is generated when UPDATE=NO and LIST=ENTRIES or ALL are specified. The valid values are:

- P Punch
- L Line printer

If this field is omitted, the default is L.

For a list of restrictions on the use of the PLI_IN command, see Chapter 9 of the DB/DC Data Dictionary Applications Guide.

PSB IN COMMAND

This command is used to access the DL/I PSB library, retrieve designated PSBs, read them into the Dictionary, and create appropriate PSB, PCB, and relationship entries. The command works in batch mode only.

The general form of the command is:

```
PSB_IN subjectname [LINKPGM|LINK|PGM] [RELATE={YES|NO}];
```

PSB_IN
PI

The long and short command verbs.

subjectname

A required operand that specifies the subject name of the PSB. For PSBs, the subject code is always P and the occurrence number is always zero. The user name is the 1-8 alphanumeric character name of the PSB in the PSB library.

LINKPGM|LINK|PGM

If the optional LINKPGM operand is specified, the PSB_IN command produces a Dictionary entry for a program whose program name is the same as the PSB name. This entry is directly connected to all PCBs, data bases, transactions, segments, and fields that are connected to the PSB. If the operand is omitted, no program entry is created.

RELATE=Y|YES|N|NO

An optional operand that specifies whether relationships with DTEs should be created. YES indicates the relationships should be created. NO indicates the relationships should not be created. The relationships created are PCB-DTE, PSB-DTE, and PGM-DTE. If this operand is omitted, the default is YES.

See Chapter 8 of the DB/DC Data Dictionary Applications Guide for a list of restrictions on the use of the PSB_IN command.

PSB OUT COMMAND

The PSB_OUT command is used to search the Dictionary and produce source statements in card images that can be used to generate PSBs. The general form of the command is:

```
PSB_OUT subjectname [DEST={P|L|T}];
```

PSB_OUT
PO

The long and short command verbs.

subjectname

A required operand that specifies the name of a PSB. The subject code is always P and the occurrence number is always zero. The user name is the 1 to 8 alphanumeric character name.

DEST=P|L|T

An optional operand that specifies the destination of the output. Omit the operand or choose one of these values:

P	punch
L	line printer
T	terminal

If the operand is omitted, the default is P.

If the information in the Dictionary is not sufficient to produce a complete set of source statements, a partial set is produced and a message is written warning the user of possible errors or omissions.

PUNCH COMMAND

The PUNCH command is used to retrieve information stored in user data segments for a subject. This command allows you to retrieve any one of the five sets of user data.

When a PUNCH command is issued, data is retrieved from the requested set of user data segments and the text portion of the data is punched. The user data line numbers are not included in the output.

The general format of the command is:

PUNCH category subjectname [U1|U2|U3|U4|U5|{UDNO=n}] + [DEST={P|L|T}];

PUNCH

The command verb.

category

This field specifies the category for which the data is to be punched.

subjectname

This field specifies the name of the subject for which the data is to be punched.

U1|U2|U3|U4|U5|UDNO=n

This operand specifies the number of the user data segment to be punched. The valid value for n is 1, 2, 3, 4, or 5. If this field is omitted, the default is user data segment 1 (UDNO=1 or U1).

DEST=P|L|T

This field specifies the destination of the output. The valid values are:

P	Punch
---	-------

L Line printer

T Terminal

If this field is omitted, the default is P.

RECALCULATE SEGMENT COMMAND

The RECALCULATE_SEGMENT command reestablishes consistency within the collection of definitions comprising a segment structure by recalculating the segment and data-group lengths, and starting positions of all fields within the segment, elements within each group, and so forth. This command calls two other commands, STRUCTURES_OUT and then either COBOL_IN or PLI_IN depending on the language parameter specified. Refer to those commands for information about restrictions and results you may expect.

The general form of the command is:

```
RECALCULATE_SEGMENT segmentname [UPDATE={YES|NO}] +  
[COMPRESS={YES|NO}] [LANGUAGE={A|B|C|J|K|L}] +  
[DEST={P|L|T}];
```

RECALCULATE_SEGMENT

RS

The long and short command verbs.

segmentname

Specifies the subject name of the segment whose structure is to be recalculated.

UPDATE=Y|YES|N|NO

Specifies whether or not the segment and element definitions are to be updated in the Dictionary as a result of the recalculation. This operand is optional; if omitted the default is NO.

YES Specifies that the definitions are to be updated.

NO Specifies that the calculated changes to the segment definitions are to be displayed for examination, but no updates are to be made.

COMPRESS=Y|YES|N|NO

Specifies how undefined areas within the segment are to be treated in the recalculation.

YES Undefined areas are to be ignored in the recalculation (that is, removed from the segment).

NO Undefined areas are to be maintained. They are to be treated as unnamed fields of the specified length.

This operand is optional; if omitted the default is NO.

LANGUAGE|LANG=A|B|C|J|K|L

Specifies the language for which the structure recalculation is to be done. The valid values are:

A	Assembler (DL/I)
B	PL/I (DL/I)
C	COBOL (DL/I)
J	Assembler (non-DL/I)
K	PL/I (non-DL/I)
L	COBOL (non-DL/I)

This operand is optional; if omitted the default is C.

DEST=P|L|T

Specifies the destination of the command output generated by the command (if UPDATE=NO). The valid values are:

P	punch
L	line printer
T	terminal (online only)

For information about and examples of the RECALCULATE_SEGMENT command see Chapter 4 of the DB/DC Data Dictionary Applications Guide.

RELOCATE COMMAND

The RELOCATE command changes the hierarchic structure of a data base or PCB by changing the sequence of its segments. The segment specified in the command is moved from its current position in the hierarchy to a new position before a specified segment or to the end of the hierarchy. You can also specify that segments subordinate to the specified segment be moved with it as a group.

Relationship data associated with the relationship between the segment and a data base or PCB is retained. The command allows you to specify a new level and parent for the segment.

The general form of the command is:

```
RELOCATE SEGMENT subjectnameA [AND DEPENDENTS] +
[IN] {DATABASE|PCB} subjectnameB [[BEFORE]SEGMENT +
segmentnameC] [PARENT=subjectnameD] [LEVEL=n];
```

**RELOCATE
RL**

The long and short command verbs.

SEGMENT

A required operand that specifies the category. Specify SEGMENT or a valid abbreviation.

subjectnameA

A required operand that specifies the subject name of the segment whose position is to be changed.

AND DEPENDENTS

An optional operand that specifies that all dependent segments are to be moved too.

IN

An optional operand.

DATABASE|PCB

A required operand that specifies whether the segment is in a data base or PCB. Specify DATABASE or PCB or any valid abbreviation.

subjectnameB

A required operand that specifies the subject name of the data base or PCB.

BEFORE SEGMENT segmentnameC

If this keyword is not specified, the relationship between the segment and data base or PCB will be established at the end of the other segments already related to the data base or PCB. If this keyword is specified, the relationship will be established before the segment specified by the segmentname field. You can use any valid abbreviation for the word SEGMENT in this operand.

PARENT=subjectnameD

An optional operand that specifies the new parent segment of the relocated segment. The operand PARENT=0 specifies no parent, which means the relocated segment is the new root segment. The subject name must be a DL/I name (language code equals A) which is a 1-8 alphanumeric character name. If this operand is omitted, the existing value of the PARENT keyword is retained.

LEVEL=n

An optional operand that specifies the new hierarchic level of the relocated segment. The value range is 1 through 15. Use only if the segment is in a data base.

LEVEL=1 and PARENT=0 mean the same thing, so both need not be specified. If you specify the PARENT name and do not specify LEVEL for a segment whose existing level is 1, the level value is changed to zero.

Only one data base or PCB hierarchy can be changed by a single RELOCATE command.

Only the segment named in the required first operand has its parent changed by PARENT through this command. Where valid, the level numbers of relocated dependents are adjusted when LEVEL is specified. They change by the same amount as the level number of the original relocated segment changes. If the relocated segment has dependents and they are not also relocated, inconsistencies in parent specifications may occur. The new parent and level specified are not checked against existing segments in the data base or PCB (as is also the case for ADD_RELATIONSHIP and CHANGE_RELATIONSHIP_DATA), so inconsistencies may also occur on this account. Other RELOCATE or CHANGE_RELATIONSHIP_DATA commands may be used to correct these problems.

If RELOCATE does not run to completion, the data base or PCB structure might be incomplete. Usually, all the data can be recovered. For a "hard" system or hardware

failure, when DL/I buffers are not written back to the data bases, THE RUN SHOULD ALWAYS BE BACKED OUT USING DL/I BACKOUT FACILITIES, because subsequent Dictionary processing against the affected data base blocks will fail. For a "soft" failure, or if backout is not possible, use the following recovery techniques:

- If the last RELOCATE message was DBD4208, no segment was partially relocated. Some desired segments may, however, not have been relocated. Obtain a report on the data base or PCB and use RELOCATE to move the rest of the segments.
- If the last message was DBD4207, a segment may have been partially relocated. Obtain a report and see if this segment is still related. If so, delete the RELOCATE dummy segment (whose name appears in the PCB or data base report). From this point, continue as in the above paragraph.
- If the segment named in the last DBD4207 message is no longer related to the data base or PCB, the data must be recovered manually. The relationship data reported under the RELOCATE dummy segment is the data previously associated with the missing segment. Use AR commands (or display form entries) to relate the missing segment to the data base or PCB again, and enter the relationship data. After the entries are made, the situation is as described in the previous paragraph. Continue by deleting the dummy segment and performing any further relocates, as described above.

REPORT COMMAND

The REPORT command creates lists of (1) the entire definition, or selected details, for a given subject; (2) printed equivalents of selected reports displayed in the interactive display forms facility; (3) a report of indirect subject references; (4) "glossary" reports of all subjects, or a selected subset of subjects, in a category with summary definitions; or (5) guide reports for installation-defined categories and relationship-types.

Subject-Specific Reports

To request all or selected details for one subject name, the form of the command is:

```
REPORT category subjectname +
[DETAIL={parameter}[(parameter),(...)]}] +
[RELATION={relationoption}[(relationoption),(...)]}] +
[RELDATA={YES|NO}] [DEST={L|T}];
```

REPORT
RPT

The long and short command verbs.

category

Specifies the category from which the report is to be obtained.

subjectname

A required operand that specifies the subject name for which the report is requested.

DETAIL=parameter

An optional operand that specifies the detail of the report. A maximum of ten options for this keyword can be specified. If this operand is omitted and a standard category has been specified, all subject data, relationship data, and relationships will be reported. If the category is installation-defined and this operand is omitted, only subject data and relationships will be reported. The valid values for this keyword when it is used with standard categories are contained in Figure 68. For more information about the keywords in this figure see Chapter 7 of the DB/DC Data Dictionary Applications Guide

The valid values for the keyword when used with installation-defined categories are:

ALL	All subject data is to be included in the report
ATTR	The attributes of the subject are to be included in the report
ALIAS	The aliases of the subject are to be included in the report
DESC	The description text of the subject is to be included in the report
EXTUSERn USERn	One of the five sets of user data text of the subject is to be included in the report (n=1 to 5)
NO	No subject data is to be included in the report

The default is ALL.

RELATION=relationoption

An optional operand that specifies which relationships of the subject are to be included in the report. Use this operand only for installation-defined categories. A maximum of ten options for this keyword can be specified. The acceptable values for relation options are:

ALL	Specifies all the relationships of the subject are to be included in the report
(* ,category)	All the relationships between the report subject and the subjects in the specified category are to be included in the report. The acceptable values for category can be any of the categories used by your installation.
(kw ,category)	All the relationships of the type identified by kw between the report subject and the subjects in the specified category are to be included in the report. The acceptable values for kw are relationship keywords appropriate to the subject named in the command. The acceptable values for category can be any of the categories used by your installation which can be related to the report subject via the specified kw.
NO	No relationships of the subject will be included in the report

The default is ALL.

Subject Data	Relationships	Relationship Data
Data Base Subjects		
DBSPATRB	DBSSEG	*SEGPATRB
DBSALLOC		*SEGPCHLD
DBSDSCR		*SEGLCHIL
DBSUSERn		*SEGLPATR
(1,2,3,4 or 5)		*SEGSRCE
	DBSPMRY	
	DBSSDRY	
	DBSDTEY	
	DBSPCB	
	DBSSYS	
	DBSEXT	
Segment Subjects		
SEGATRB	SEGPMRY	
SGPLINFO	SEGSDRY	
XFLD1	SEGDBS	
XFLD2	SEGDE	
XFLD3	SEGPCB	
SEGDSCR	SEGSYS	
SEGUSERn	SEGEXT	
(1,2,3,4, or 5)		
Element Subjects		
DTEATRB	DTEPMRY	
DTECATRB	DTESDRY	
DTEVALUE	DTECNTNS (subordinate elements)	
DTELOCC	DTECNTNW (superior elements)	
DTEPATRB	DTEDBS	
DTEEDIT	DTESEG	
PLIINFO	DTEPCB	
DTEDSCR	DTESYS	
DTEUSERn	DTEEXT	
(1,2,3,4, or 5)		
PCB Subjects		
PCBDATRB	PCBSEG	*SNSGINDC
PCBTATRB	PCBPMRY	
PCBDSCR	PCBSDRY	
PCBUSERN	PCBDBS	
(1,2,3,4, or 5)	PCBDTERB	
	PCBSYS	
	PCBEXT	
System Subjects		
SYSATRB	SYSSYS	
SYSDSCR	SYSDBS	
SYSUSERn	SYSSEG	
(1,2,3,4, or 5)	SYSDE	
	SYSPCB	
	SYSEXT	

* Any command specifying the relationship data segments for DBS-SEG or PCB-SEG relationships must also include in the `DETAIL=` operand the relationship segment `DBSSEG` or `PCBSEG`.

Figure 68. Valid `DETAIL` parameters

RELDATA=YES|NO

An optional operand that specifies whether the attributes for each relationship specified by the `RELATION` keyword will be included in the report. Use this operand only for installation-defined categories. The default is `NO`.

DEST=L|T

An optional operand that specifies the destination of the output for online systems. L specifies line printer output and T specifies terminal output. If this operand is omitted terminal output is assumed. Printer output is assumed for batch operation.

Display Form Equivalent Reports

To request printed equivalents of the interactive displays, the command is:

```
REPORT category subjectname {NAME|RPTNAME}=reportname +
[DEST={L|T}];
```

**REPORT
RPT**

The long and short command verbs.

category

Specifies the category from which the report is to be obtained.

subjectname

A required operand that specifies the subject name for which the report is requested.

NAME|RPTNAME=reportname

A required operand that specifies the type of report. See Figure 69. For examples of these reports and information about how to request them, see Chapter 7 of the DB/DC Data Dictionary Applications Guide.

DEST=L|T

An optional operand that specifies the destination of the output for online systems. L specifies line printer output and T specifies terminal output. If this operand is omitted terminal output is assumed. Printer output is assumed for batch operation.

Indirect Subject Reference Reports

To request a report showing indirect references to a data base, field, or segment name, use this form of the command:

```
REPORT category subjectname {NAME|RPTNAME}=ISR +
[DETAIL={parameter|((parameter),(...))}] [DEST={L|T}];
```

**REPORT
RPT**

The long and short command verbs.

category

A required operand that specifies the category from which the report is to be obtained. Valid categories are ELEMENT, SEGMENT, and DATABASE or their valid abbreviations, only.

Report Names Category

General Reports:

ALIAS ELEMENT
 SEGMENT
 DATABASE
 PCB
 installation-defined categories

DSCR or DESC Any

USER or USERSEG Any

PLI or PLIDATA ELEMENT
 SEGMENT

ISR ELEMENT
 SEGMENT
 DATABASE

Element Reports: ELEMENT

DTE
FLDSEG
SUBELT
SUPELT

Segment Reports: SEGMENT

SEG
SEGFLD

Database Reports: DATABASE

DBS
HIER
STRUC
PRINDX
SECINDX

PCB Reports: PCB

PCB
SENSEG

System Reports:

SDF* SYSDEF
SYS* SYSTEM
JOB* JOB
MODULE* MODULE
PGM or PROGRAM PROGRAM
TRAN TRANSACTION
PSB PSB
SDFPSB SYSDEF
SDFDBS SYSDEF
SDFTRN SYSDEF
PSBTRN PSB

* The report for SDF, SYS, JOB or MODULE is actually a report of the description entries for the specified subject.

Figure 69. Valid Report Names for NAME=reportname

subjectname

A required operand that specifies the subject name for which the report is requested.

NAME|RPTNAME=ISR

This operand indicates an indirect subject reference report is desired.

DETAIL=parameter

This optional operand specifies the fields to be searched for if NAME=ISR is specified (that is, if an indirect subject reference report is requested). A maximum of ten parameters can be specified. The acceptable parameters for this keyword depend on the category specified. If this operand is omitted, all relevant fields will be searched.

If the category is ELEMENT the acceptable values are:

SRCHNAME	DDATA
SUBSEQ	INDXFLNM

If the category is SEGMENT the acceptable values are:

ISSNAME	PAIRNAME
LCSGNAME	PCSGNAME
LPSGNAME	PHYPAR
PARENT	SRCSNAME

If the category is DATABASE the acceptable values are:

INDCNAME	PROCSEQ
LCDBNAME	SRCDNAME
LPDBNAME	

DEST=L|T

An optional operand that specifies the destination of the output for online systems. L specifies line printer output and T specifies terminal output. If this operand is omitted terminal output is assumed. Printer output is assumed for batch operation.

Glossary Reports

To request a report of all subjects, or a selected subset of subjects, in a category, with a summary of their characteristics, use this form of the REPORT command:

```
REPORT GLOS category [NAME=string] +
[STAT={status|(status,...)}] [CODE={code|(code,...)}] +
[OCCUR={occurrence|(occurrence,...)}] +
[FROM=username1] [TO=username2] +
[SORT={NO|sortfield|(sortfield1,sortfield2,...)}] +
[DESC={1|n}] [ATTR={YES|NO}] +
[DEST={L|T}];
```

**REPORT
RPT**

The long and short command verbs.

GLOS Specifies a glossary report.

category Specifies the category from which the report is to be obtained. *SYO may be specified for a report from all system categories.

NAME=string An optional operand that specifies that only subject names containing the specified string are to be reported. The value can be 1-31 characters long. If the character string contains blanks or special characters, enclose it in single quotation marks. Refer to "Character Strings" under the SCAN command for an explanation of characters which have special meanings when used in this string.

STAT=status|(status, ...) An optional operand that specifies the status codes to be used in selecting subjects. The acceptable status codes for this keyword are 0 to 9 and A to T. A maximum of ten status codes can be specified.

CODE=code|(code, ...) An optional operand that specifies the subject codes to be used in selecting subjects. One to ten subject codes can be specified. The acceptable values depend on the category specified. The table below shows the subject codes that can be used for a category.

CATEGORY	SUBJECT CODE
*SYO	G, J, M, N, P, S, T, and U
SYS	S
SDF	N
JOB	J
PGM	G
MOD	M
TRN	T
PSB,PCB	P
DBS	F, L, P, X, and Y
SEG,DTE	A, B, C, J, K, and L

This operand is ignored if the category specified is installation-defined.

OCCUR=occurrence|(occurrence, ...) An optional operand that specifies the occurrence numbers to be used in selecting subjects. A maximum of ten occurrence numbers can be specified. The acceptable values for this keyword are 0 to 255.

FROM=username1 An optional operand used for setting a lower limit for user names to be reported. Each name retrieved is compared with this value. Only those names that occur in collating order at or after username1 are reported. The value of username1 can be 1 to 31 characters long. If the character string contains blanks or special characters, enclose it in single quotation marks.

TO=username2

An optional operand used for setting an upper limit for user names to be reported. Each name retrieved is compared to this value. Only those names that occur in collating order at or before username2 are reported. The value can be 1-31 alphameric characters long. If the value contains blanks or special characters, enclose it in single quotation marks.

SORT=NO|sortfield|(sortfield1, sortfield2...)

This operand specifies whether the subject names should be printed in collating sequence. The sortfields specify which name or name qualifiers are to be used for sorting. If NO is specified, no sorting is done. The valid values for sortfield are:

STAT specifies the status code
CODE specifies the language or subject code
NAME specifies the user name
OCC specifies the occurrence number

Up to four sort fields can be specified with the command. The first operand specified will be the major sort field. Any subsequent fields will be minor sort fields.

This operand is only valid in batch mode. The default is NAME.

DESC=1|n

This operand specifies the maximum number of lines of description text that should appear in the report. The value can be any number in range 0 through 255 inclusive. The default is 1.

ATTR=Y|YES|NO

This operand specifies whether the attributes are to be included in the summary report. The default is YES.

DEST=L|T

An optional operand that specifies destination of the output for online systems. L specifies line printer output and T specifies terminal output. If this operand is omitted terminal output is assumed. Printer output is assumed for batch operation.

Guide Reports

To request reports for installation-defined categories or relationship types, use this form of the command:

```
REPORT {CATEGORY|RELTYPE} subjectname NAME=GUIDE +
[DEST={L|T}];
```

**REPORT
RPT**

The long and short command verbs.

CAT|CATEGORY|RELTYPE

Specifies an installation-defined category or a relationship type.

subjectname

A required operand that specifies the subject name for which the report is requested.

NAME=GUIDE

A required operand that specifies a guide report. The guide report contains information about an installation-defined category or relationship type and its associated attributes. This report provides you with the information you need to define subjects in the category or relationships of the type specified in this command.

DEST=L|T

An optional operand that specifies destination of the output for online systems. L specifies line printer output and T specifies terminal output. If this operand is omitted terminal output is assumed. Printer output is assumed for batch operation.

SCAN COMMAND

This command scans subject names, descriptions, user data, or relationships in a specified category to find one or two specified character strings. There are three forms of the command; the first scans subject names only, the second scans descriptions, User Data, descriptions and subject names, or User Data and subject names, and the third scans relationships or relationships and subject names. Results are reported in summary or in detail as specified in the command.

The form of the command for scanning subject names is:

```
SCAN category [NAME=string] [STAT={status|(status,...)}] +
[CODE={code|(code,...)}] +
[OCCUR={occurrence|(occurrence,...)}] [FROM=username1] +
[TO=username2] [START=subjname] [LST|TXT|RPT] +
[{{LIMIT|LM}=n} [DEST={L|T}];
```

The form of the command for scanning descriptions, User Data, descriptions and subject names, or User Data and subject names is:

```
SCAN category {{DESC|Un}=string [{{AND|OR|ANDNOT} +
string2} . [LINENO=n|(min,max)] [NAME=string] +
[STAT={status|(status,...)}] [CODE={code|(code,...)}] +
[OCCUR={occurrence|(occurrence,...)}] + [FROM=username1] +
[TO=username2] [START=subjname] [LST|TXT|RPT] +
[{{LIMIT|LM}=n} [DEST={L|T}];
```

The form of the command for scanning relationships or relationships and subject names is:

The operands for all forms of the command are described below:

SCAN
SC

The long and short command verbs.

```
SCAN category [REL=(reltype,string)] +
[NAME=string] [STAT={status|(status,...)}] +
[CODE={code|(code,...)}] +
[OCCUR={occurrence|(occurrence,...)}] [FROM=username1] +
[TO=username2] [START=subjname] [LST|TXT|RPT] +
[LIMIT|LM]=n [DEST={L|T}];
```

category

An required first operand that specifies the category to be scanned.

DESC|Un

A required operand to scan text data for each subject in the category. Choose one of the following keywords:

DESC all description text

Un all User Data (U) in the specified set of User Data (1 to 5)

string1

Specifies the character string that the scan looks for. The value can be 1 to 31 characters long. If the string contains blanks or special characters, enclose it in single quotation marks. SCAN will treat all text as if it were a continuous character string, so the string may be found as a result of concatenating two successive text lines. Refer to Chapter 7 of the DB/DC Data Dictionary Applications Guide for information about how SCAN works. See "Character Strings" below for more information.

AND|OR|ANDNOT

This operand may only be used with DESC|Un and must immediately follow that operand. It specifies the boolean operator prefix for string2. Required if string2 is specified. Choose one of the following:

AND find string1 and string2 in the same operand

OR find string1 or string2 or both

ANDNOT find string1 in the field with no string2

string2

Specifies the second character string that the scan looks for (DESC and USER scans only). The value is 1-31 characters. If the string contains blanks or special characters, enclose it in single quotation marks. See "Character Strings" below for more information.

LINENO=n|(min,max)

An optional operand that specifies the line numbers to be scanned. The valid values are:

n specifies a line number

(min,max) specifies an inclusive set of line numbers

If this operand is omitted, the default is a scan of every line of the type of text specified for each subject in the specified data base.

REL=(relob, string)

This operand specifies that the names of related subjects in another category, designated by relob, are to be scanned for the string.

relob The relob parameter is a three-letter abbreviation for the standard category:

DTE	PGM
SEG	JOB
DBS	SYS
PCB	SDF
TRN	PSB
MOD	

For relationships between two elements there are two additional relobs:

CNTNS Contains

CNTNW Is contained within

The value of relob can be the same as that of category (the first operand in the command). For DBS, PCB, DTE, and SEG, the result is a scan of the names.

string Specifies the character string that the scan looks for. The value can be 1-31 characters long. If the string contains blanks or special characters, enclose it in single quotation marks. See "Character Strings" below for more information.

NAME=string

Specifies that all primary names and aliases specified by **string** are to be scanned. The value can be 1 to 31 characters long. If the string contains blanks or special characters, enclose it in single quotation marks. See "Character Strings" below for more information.

STAT=status[(status, ...)]

An optional operand that specifies the status codes to be used in selecting subjects. The acceptable status codes for this keyword are 0 to 9 and A to T. A maximum of ten status codes can be specified.

CODE=code[(code, ...)]

An optional operand that specifies the subject codes to be used in selecting subjects. One to ten subject codes can be specified. The acceptable values depend on the category specified. The table below shows the subject codes that can be scanned for a category.

CATEGORY	SUBJECT CODE
SYS	S
SDF	N
JOB	J
PGM	G
MOD	M
TRN	T
PSB,PCB	P
DBS	F, L, P, X, and Y
SEG,DTE	A, B, C, J, K, and L

This operand is ignored when the category specified is installation-defined.

OCCUR=occurrence|(occurrence, ...)

An optional operand that specifies the occurrence numbers to be used in selecting subjects. A maximum of ten occurrence numbers can be specified. The acceptable values for this keyword are 0 to 255.

FROM=username1

An optional operand used for setting a lower limit for user names to be reported. Each name retrieved is compared with this value. Only those names that occur in collating order at or after username1 are reported. The value of username1 can be 1 to 31 characters long. If the character string contains blanks or special characters, enclose it in single quotation marks.

TO=username2

An optional operand used for setting an upper limit for user names to be reported. Each name retrieved is compared to this value. Only those names that occur in collating order at or before username2 are reported. The value can be 1 to 31 characters long. If the value contains blanks or special characters, enclose it in single quotation marks.

START=subjname

An optional operand that specifies the starting position of the scan in the specified data base. The acceptable value for subjname can be any fully-qualified subject name which exists in the data base. If this operand is omitted, the starting position of the scan occurs with the first subject name in the data base. If the subject specified has not been defined in the Dictionary, you will receive an error message.

LST|TXT|RPT

Specifies the scope of the output. Choose one of the permitted values or omit:

LST	Lists names of subjects meeting criteria.
TXT	Lists names and reports subjects meeting criteria. If names or relationships are scanned, a full report is produced. If User Data or descriptions are scanned, only text data meeting

criteria is produced in the report.

RPT Lists names and produces full report on subjects meeting criteria.

The default is LST. TXT is meaningless for the scan of NAME; the result is the same as RPT.

LIMIT|LM=n

An optional operand that specifies that scanning is to stop after n hits. Use this operand when RPT or TXT has been specified. The value can be from 1 to the limit defined by your installation. If no upper limit has been set it is 32767. If this operand is omitted, the default is limit is used. This limit may be set by your installation. If your installation has not specified a default value, the value is 10.

DEST=L|T

An optional operand that specifies the destination of RPT or TXT option output. The valid values are:

L	Printer
T	Terminal

The online default is L.

Character Strings

Blanks in scan strings must match blanks in field searched for a hit. String1 and string2 may contain special characters, but two characters have a special meaning.

/
Matches any character in field being scanned. Ignored as leading character. Thus if ABCARDCO is in the field scanned, both "//ABC" and "ABC//CO" will result in hits.

<
In first position, means the following string must appear in first position for a hit. In a scan of User Data or Description text, it is ignored.

See Chapter 9 of the DB/DC Data Dictionary Applications Guide for information about and restrictions on the use of the SCAN command.

THE SET COMMANDS

These commands, except for SETCKPT, SETLOGU, and SETLOGI, temporarily change default values for the subject name qualifiers. The temporary defaults set are in effect until changed or for the duration of the batch run. When the run is completed, the system-wide defaults are restored.

The command to change the default status qualifier is:

SETSTAT={A-T|0-9}

where P is used for production status.

There are two commands to change the subject code default. The first is:

SETLANG={A|B|C+ |J|K|L}

where A, B, C, J, K, and L are the subject codes that identify the programming language of a segment or element:

A=assembler language, DL/I

B=PL/I, DL/I

C=COBOL, DL/I

J=assembler language, non-DL/I

K=PL/I, non-DL/I

L=COBOL, non-DL/I

The SETLANG Command has no effect on the COBOL_IN and PLI_IN commands. Both these commands have a LANGUAGE operand which takes precedence over the SETLANG command. The value specified for the LANGUAGE operand, or the default value inherent to the command is always used. See the individual commands above for more information.

The second sets the subject code default for the data base subject category:

SETDBTP={P|L|F+ |X|Y}

where P, L, F, X, and Y are data base types:

P=physical data base (DL/I)

L=logical data base (DL/I)

F=non-DL/I data set (file)

X=primary index data base (DL/I)

Y=secondary index data base (DL/I)

There is no SET command for the other categories (PCB, SYSDEF, system, job, program, module, transaction, PSB, and installation-defined) because they each have just one possible subject code default value.

The SET command to change the occurrence number default is:

SETOCR=n

where n is any number from 0 to 255.

The default occurrence does not apply to PCB names, as explained in the discussion of subject name qualifiers above, or to PSB names, for which a zero occurrence is required.

The SET command to specify the frequency that a DL/I checkpoint is to be taken by the Dictionary is:

SETCKPT=s

where s is the frequency in elapsed seconds.

The SET checkpoint command can be used to request that a DL/I checkpoint be taken automatically at a specified frequency, or to discontinue such a request. The command is valid for batch Dictionary execution and online operation under CICS/VS. The value of s can be in the range 0 to 255. A value of 0 specifies that no checkpoints are to be initiated automatically.

The frequency set becomes a target for the Dictionary, so that on initiation or completion of each individual command in the user input, the Dictionary can examine the time that has elapsed since the last checkpoint was taken (or start of the execution), and if the number of seconds is sufficient, invoke a DL/I checkpoint. An additional criterion is that Dictionary data base modifications may have been made during the elapsed period of time.

The SET command that controls logging of internally generated commands is:

SETLOGI=Y|N for yes and no

This command causes all commands generated internally to be logged in a file you may print.

The SET command that controls logging of user generated commands is:

SETLOGU=Y|N for yes and no

This command causes all commands that you enter to be logged in a file you may print. SETLOGU=N should never be specified when running in batch, since the log output is the normal batch output.

Note: If LOGI or LOGU has been set to YES in the default table, then the corresponding SETLOG command will not have any affect. Specifying SETLOG=N or SETLOGU=N will generate an error message.

SIGN ON COMMAND

The SIGN_ON command is used to:

- Identify a user to the Dictionary
- Authorize processing according to the status codes and categories available to that user.

The checking of the user identification and password always takes place after this command is entered. If your installation has enabled Dictionary security, then security checks will be applied on each subsequent command. When security has not been enabled security checking of subsequent commands is not performed.

The general format of the command is:

```
SIGN_ON ID=username PASSWORD=password ;
```

SIGN_ON

The command verb.

ID=username

A required operand that signifies the user identification. This has to be an existing subject in the DDUSER category. The identification can be from 1 to 31 characters in length.

PASSWORD=password

A required operand that signifies the password uniquely associated with the user identification. The password can be from 1 to 8 characters in length, but cannot be all blanks.

If security is enabled, this command must be the first command entered in a batch input stream for any user. The only commands that may precede the SIGN_ON command are the SET commands (except SETCKPT) and the FLUSH command. If you are using CMNDMOD or any other online entry MOD, and security is enabled, you must also enter the SIGN_ON command first. If you are using the display forms, security sign-on is handled by the security HEADER form.

If for some reason the command identification is found to be in error, all input following the command, including batch form and report requests, will be flushed until the next SIGN_ON command is encountered. In this event, syntax checking will take place for the flushed commands.

STAGE 1 OUT COMMAND

The STAGE_1_OUT command is used to produce source statements in card image form for IMS/VIS system definition. This information is for the DATABASE, APPLCTN, and TRANSACT macros.

The STAGE_1_OUT command offers the option of punching the system definition information into cards that can be inserted into the IMS/VIS Stage-1 job stream.

An option of the command allows specification of the IMS/VIS level that you are using.

The general format of the command is:

```
STAGE_1_OUT subjectname [IMSLVL={1.1.3|1.1.4|1.1.5}] +  
[DEST={P|T|L}];
```

STAGE_1_OUT S10

The long and short command verbs.

subjectname

This operand specifies the Dictionary name of the system definition subject for which the system definition macros are to be produced.

IMSLVL=1.1.3|1.1.4|1.1.5

This operand specifies the IMS/VIS level for which system definition is required. The valid values are:

- 1.1.3 Specifies release 1.1.3
- 1.1.4 Specifies release 1.1.4
- 1.1.5 Specifies release 1.1.5 and later

If this field is omitted, the default is 1.1.4.

DEST=P|L|T

This operand specifies the destination of the output of the command. The valid values are:

- P Punch
- L Line printer
- T Terminal

If this field is omitted, the default is P.

STRUCTURES OUT COMMAND

This command searches the segment and element categories of the Dictionary and generates programming language data structures in COBOL, PL/I, or assembler language. A single command produces one member, including IEBUPDTE (for OS/VS) or MAINT (for DOS/VS) control cards, for use in updating the copy or include library which should contain the member. You may produce multiple segment structures at one time either by specifying the names of the desired segments in one command or by naming a higher-level Dictionary subject to which the segments are related.

The STRUCTURES_OUT command can also be used as a report medium by directing the output to a printer. You can request additional information in the form of embedded comments in the output detailing the Dictionary definitions and the parameter values underlying the output structure. As a further aid in pre-modification analysis, you can request that STRUCTURES_OUT delete all undefined areas (normally produced as fillers) from the output structure.

The general form of the command is:

```
STRUCTURES_OUT category {subjectname|(subjectname,...)} +
[MEMBER=name] [LANG={A|B|C|J|K|L}] +
[DVALUES={YES|NO}] [COMPRESS={YES|NO}] +
[SEGATR={YES|NO|data-string}] [LEVEL={1|n}] +
[LVL TWO=value] [LVL INCR=n] [NARATIV={YES|NO}] +
[PFX=value] [SFX=value] [DEST={P|L|T}];
```

STRUCTURES_OUT

SO

The long and short command verbs.

category

A required operand that specifies the category of the subject to be retrieved. The only valid categories are SEGMENT, DATABASE, PCB, PROGRAM, or MODULE (long or abbreviated forms).

subjectname

A required operand that specifies the subject name of the subject to be retrieved. If the category is SEG, more than one subject name may be specified. If more than one subject name is specified, each name must be enclosed in parentheses and separated by commas, and the whole list must be enclosed in a second set of parentheses. Multiple subject names should be in hierarchic sequence if they are being used for DL/I path calls. Four-part subject names and user names can be mixed in a command. The user names must also be in parentheses, as in this example:

```
SO SEG ((P,A,RPTSEG,0),(COBSEG),(SEG4)) LANG=A +
MEMBER=S1;
```

At most 15 full four-part subject names can be included in a single command.

MEMBER|MEM=name

An optional operand that specifies the library member name to be assigned to the data structure. If this operand is omitted, NONAME is assumed.

LANGUAGE|LANG=A|B|C|J|K|L

An optional operand that specifies the programming language in which the structure is to be built. If the operand is omitted, A is the default. Possible values are:

A	Assembler (DL/I)
B	PL/I (DL/I)
C	COBOL (DL/I)
J	Assembler (non-DL/I)
K	PL/I (non-DL/I)
L	COBOL (non-DL/I)

DVALUES|DVALS=Y|YES|N|NO

This operand specifies whether the Dictionary values are to be placed as comments in the structure. YES or Y specifies that the values are to be included. NO or N specifies that they are not to be included. If this field is omitted, the default is N.

COMPRESS=Y|YES|N|NO

This operand specifies whether undefined areas should be deleted from the output structure. YES or Y specifies that the undefined areas are to be deleted. NO or N specifies that they are to be retained. If this field is omitted, the default is N.

SEGATR=Y|YES|N|NO|data-string

An optional operand for PL/I structures that specifies that the scope and storage class attributes are to be appended to the segment level. YES specifies that the attributes specified in the default table are to be appended. NO specifies that the attributes are not to be appended. Data string specifies a character string of attributes or text is to be appended. The default is YES.

LEVEL|LVL=1|n

An optional operand for COBOL or PL/I structures that specifies the initial level number for segments. 1 specifies the level number 1 for each segment. The value n specifies a level number other than 1 for each segment. Valid values are 0-255. If the operand is omitted, the default is 1.

Use of the value n forces the value of SEGATR to default to NO.

LVL TWO=value

An optional operand for COBOL or PL/I structures that specifies the level number that STRUCTURES_OUT will assign to second level fields. Valid values are 2 to 10. If this operand is not specified the value in the default table will be used.

LVLINCR=n

An optional operand for COBOL and PL/I structures that specifies the increment to be added by STRUCTURES_OUT to calculate the level number for third and subsequent level fields. Valid values are 1 to 10. If this value is not specified, the value specified by your installation in the default table will be used.

NARATIV|NTV=Y|YES|N|NO

An optional operand that specifies whether narrative (PLIINFO where PLI TYPE=N) can precede segment and element declarations. YES specifies that the narrative should be produced. NO specifies that the narrative should not be produced. If the operand is omitted, the default is YES.

PFX|PREFIX=value

An optional operand that specifies a prefix of 1 to 8 characters for all items appearing in the structure. If the operand is omitted, no default is assumed.

SFX|SUFFIX=value

An optional operand that specifies a suffix of 1 to 8 characters for all items appearing in the structure. If the operand is omitted, no default is assumed.

DEST=P|L|T

An optional operand that specifies the output device. The values for the code are:

P	Punch
L	Line printer
T	Terminal

During STRUCTURES_OUT processing, any comments added in the PLIDATA with PLITYPE=C segment will be produced in the output. (See the ADD command in this chapter for the method of adding comments.) See Chapter 9 of the DB/DC Data Dictionary Applications Guide for more information about and examples of this command.

APPENDIX A: DICTIONARY KEYWORDS FOR SUBJECT DATA

This appendix defines all the keywords used to specify subject data for subjects in standard categories. All the keywords listed are used with the ADD command and the CHANGE_IN command. The keyword definitions are presented in a series of tables. All of the keywords in the tables are germane to subjects in standard subject categories. These keywords may be associated with a given subject name independent of its relationships with other subjects.

The major division of the tables is according to a Dictionary physical data base. Definitions are grouped for the DATABASE, ELEMENT, PCB, SEGMENT, and SYSTEM data collections. One or more tables also represent a further subdivision of the data and are given a title that summarizes the kind of data manipulated by the keywords grouped in that table, or its continuation.

The collection of data identified by the subheading of each table is stored in a segment of a Dictionary data base. The segment names are given for use with the DELETE_DATA and REPORT commands.

Entire groups of data can be deleted with the DELETE_DATA command by specifying the segment name as a value for the DATA= operand. One or more segments, each containing the data associated with a grouping of keywords, can be deleted by the action of the DELETE_DATA command.

The segment names are also used to specify a group of data in a report. One or more segment names can be specified for the DETAIL= operand of the REPORT command.

Users of display forms referring to these tables should understand the different requirements for subject names and differences in free-form text entries in the display forms.

Notes:

1. The first column of each table gives the keyword name and alternative names; the second gives examples of its use; and the last three columns give the length and type of entry, allowable values, and a description of the meaning and use of the keyword.
2. In the Length and Type column, A=alphabetic, N=numeric, and A/N=alphameric.
3. An entry in the Description column noting that a keyword is DL/I-compatible always refers to the first keyword in column 1.
4. "Not edited" means that only length and type are validated.
5. Required ID keywords are listed at the top of the page.

DATABASE

PHYSICAL ATTRIBUTES OF THE DATA BASE

No ID keyword required.
Segment name for DELETE_DATA and REPORT commands: DBSPATRB

Keyword	Example	Length /Type	Allowable Values	Description
ACCESS DBACCS DBSACCS	DBACCS=HDAM ACCESS=(HISAM, OSAM) ACCESS=(INDEX, VSAM,PROT, DOSCOMP) ACCESS=(INDEX, VSAM,NOPROT) ACCESS=BPAM DBACCS=(INDEX, VSAM,,DOSCOMP)	4-6A 4A 4 6A 7A	One-four values: DL/I or OS access method Supporting access method for DL/I data base PROT or NOPROT (INDEX, VSAM only) DOSCOMP (INDEX, VSAM only) Not edited.	Access method for this data base or file. May be substituted for Dictionary keywords OSACCS, PROT, and DOSCOMP. Only one value required for non-DL/I file documentation. First value required for DBD_OUT. DL/I compatible keyword.
INDACCS OSACCS	INDACCS=OSAM OSACCS=ISAM	4A	BSAM,ISAM, OSAM,VSAM Not edited.	Supporting access method used. See ACCESS. Optional for DBD_OUT.
PROT	PROT=P PROT=N	1A	P=PROT N=NOPROT	Indicates whether PROT option is specified (INDEX,VSAM only). See ACCESS. Optional for DBD_OUT.
DOSCOMP	DOSCOMP=D DOSCOMP='	1A	D=DOSCOMP specified ' '=DOSCOMP not specified	Indicates whether DOS compatibility option is specified for this data base (INDEX, VSAM only). Optional for DBD_OUT.
RMNAME	RMNAME= (MOD,10,100,300)	1-8A 1-3N 1-8N 1-8N	Four values: (MOD) (ANCH) 0-255 (RBN) 0-16777215 (BYTES) 0-16777215	Specifies data about an HDAM data base. May substitute for dictionary keywords MOD, ANCH, RBN, BYTES. All values must be entered when keyword is specified. See <u>IMS/VS Utilities Reference Manual</u> . Optional for DBD_OUT. DL/I-compatible keyword.
MOD DBRMNM DBSRMNM	MOD=RTNRMI DBRMNM=RANMOD	1-8A/N	Not edited.	Name of user-supplied randomizing module used for storing and accessing segments in an HDAM data base. Optional for DBD_OUT.

Keyword	Example	Length /Type	Allowable Values	Description
ANCH DBRAPNO DBSRAPNO	ANCH=25 DBRAPNO=1	1-3N	0-255	Number of direct access anchor points in each physical block or QSAM data set used in primary data set group of an HDAM data base. Optional for DBD_OUT.
RBN DBMAXRB DBSMAXRB	RBN=7350 DBMAXRB=100	1-8N	0-16777215	Maximum QSAM relative block number value the randomizing module is allowed to produce for this data set (HDAM only). Optional for DBD_OUT.
BYTES DBBYTES DBSBYTES	BYTES=350 DBBYTES=3625	1-8N	0-16777215	Maximum number bytes of a data base record that may be inserted into DBS root segment addressable area in a single sequence of inserts (HDAM only). Optional for DBD_OUT.
PASSWD	PASSWD=YES	1-3A	Y YES N NO	Specifies that DBDNAME is used as VSAM password.
QSEC MINFQS	QSEC=10 MINFQS=101	1-3A/N	Not edited	Specifies an access code for Query access security.
USEC MINFUS	USEC=10 MINFUS=101	1-3A/N	Not edited	Specifies an access code for update access security.
DATM DATMUSER	DATM=(1,10) DATMUSER=(2,100)	1N 1-3N	Two values: 1. 1-5 2. 1-999	Specifies the User Data segment and the starting sequence number for the DATM/JCL statements. The end of such statements is indicated by an ENDDATA statement.
DATMNUM	DATMNUM=1	1N	1-5	Specifies the User Data segment for storing DATM/JCL statements.
DATMOCC	DATMOCC=10	1-3N	1-999	Specifies the starting sequence number for DATM/JCL statements in a User Data segment.

PHYSICAL CHARACTERISTICS OF DATA SETS

ID keyword required: DSGNMBR=1-10
 Segment name for DELETE_DATA and REPORT commands: DBSALLOC

Keyword	Example	Length /Type	Allowable Values	Description
DSGNMBR DSG	DSGNMBR=4 DSG=10	1-2N	1-10	Data set group number for which these characteristics are applicable. Required for DBD_OUT.
DSGLABEL LBL	LBL=DSG1 DSGLABEL=SEGDD	1-8A/N	Not edited.	Identifying label coded on data set statement for this data set group. Optional for DBD_OUT. <u>ID keyword required.</u>
DD1 PRIMDS DD1NAME	DD1=DDSDTE PRIMDS=DS1	1-8A/N	Not edited.	Ddname of primary data set of this data group. Required for DBD_OUT. DL/I compatible keyword. <u>ID keyword required.</u>
DD2 OUTDS DD2NAME	DD2=DDSODTE OUTDS=OUTPUTDS	1-8A/N	Not edited.	Ddname of the output data set used when ACCESS=HSAM, BSAM, QSAM. Optional for DBD_OUT. DL/I-compatible keyword. <u>ID keyword required.</u>
BLKSIZE1 SIZE1	BLKSIZE1=7294 SIZE1=3440	1-5N	0-32767	Blocksize for this data set group. Optional for DBD_OUT. <u>ID keyword required.</u>
RCDSIZE1	RCDSIZE1=80	1-5N	0-32767	Data management logical record size for this data set group. Optional for DBD_OUT. <u>ID keyword required.</u>
BLKSIZE2 SIZE2	BLKSIZE2=800 SIZE2=6900	1-5N	0-32767	Blocksize for this data set group. Optional for DBD_OUT. <u>ID keyword required.</u>
RCDSIZE2	RCDSIZE2=132	1-5N	0-32767	Data management logical record size for this data group. Optional for DBD_OUT. <u>ID keyword required.</u>

Keyword	Example	Length /Type	Allowable Values	Description
OVFLW OSAMDS DD3NAME	OVFLW=RSJDD2 OSAMDS=OVERFLOW	1-8A/N	Not edited.	Ddname of overflow data set of this data set group when ACCESS=INDEX, HISAM. Optional for DBD_OUT. DL/I compatible keyword. <u>ID keyword required.</u>
DEVICE DEVTYPE DEVNAME	DEVICE=3330 DEVTYPE=2400	4N	Typical values include: 2314, 2319, 2305, 3330, 3340, 3350, 2400, 3400, and TAPE. Not edited.	Type of physical storage device on which all data sets for this data set group are to be stored. Required for DBD_OUT. DL/I compatible keyword. <u>ID keyword required.</u>
MODEL	MODEL=11 MODEL=2	1-2N	1=Model of 2305 or 3330 2=Model of 2305 11=Model of 3330	Model of 2305 or 3330 device. Optional for DBD_OUT. DL/I-compatible keyword. <u>ID keyword required.</u>
SCAN SCANLMT	SCAN=7 SCANLMT=0	1-3N	0-255	Number of DASD cylinders to be scanned when searching for storage space during segment insertion (HDAM, HIDAM). Optional for DBD_OUT. DL/I compatible keyword. <u>ID keyword required.</u>
FRSPC	FRSPC=(10,2) FRSPC=(0,21)	1-3N 1-2N	Two values: 1. 0-100, but not 1 2. 0-99 Not edited.	Information about free space left in DS when loading. Allows specification of FRSPCBF (first value) and FRSPCSP in one keyword. Optional for DBD_OUT. DL/I compatible keyword. <u>ID keyword required.</u>
FRSPCBF	FRSPCBF=10	1-3N	0-100, but not 1 Not edited.	Free block frequency factor that specifies every nth block in this data set group will be left as freespace (HDAM or HIDAM). Optional for DBD_OUT. <u>ID keyword required.</u>

Keyword	Example	Length /Type	Allowable Values	Description
FRSPCSP	FRSPCSP=2	1-2N	0-99 Not edited.	Percent factor that specifies minimum of each block to be left as free space (HDAM or HIDAM). Optional for DBD_OUT. <u>ID keyword required.</u>
RECFM	RECFM=FB RECFM=VBA	1-5A	Not edited.	Specifies record format for OS files and GSAM data bases (IMS/VS 1.1 and higher levels.) See <u>JCL Reference Manual</u> for valid values. Optional for DBD_OUT. <u>ID keyword required.</u>

FREE-FORM DESCRIPTION

ID keyword required: TEXTSEQ=1-999
 Segment name for DELETE_DATA and REPORT commands: DBSDSCR

Keyword	Example	Length /Type	Allowable Values	Description
DESC DESCRIPTION DBSDSCR	DESC=(3,'TEXT INFO') DESCRIPTION= (1,'HDR')	1-3N 1-72 A/N	Two values: 1. n=line number, 1-999 2. Free-form text data. To include special characters, enclose in single quotes. Not edited.	Free-form text describing data base.
TEXTSEQ	TEXTSEQ=3	1-3N	1-999	Line number for TEXT. Same as first parameter of DESCRIPTION keyword.
TEXT	TEXT=('PHYS DBS DATABASE')	1-72 A/N	Free-form text. To include special characters, enclose in single quotes. Not edited.	Same as second parameter of DESCRIPTION keyword. <u>ID keyword required.</u>

USER DATA

No ID keyword required.
 Segment name for DELETE_DATA command: DBSUSERn; n=1-5

Keyword	Example	Length /Type	Allowable Values	Description
DBSUSER1 DBSUSER2 DBSUSER3 DBSUSER4 DBSUSER5	DBSUSER1= (1,'TEXT') DBSUSER2=(7,'PGM NAME') DBSUSER5=(1,'')	1-3N 1-80 A/N	Two values: 1. n=line number, 1-999 2. Free-form text data. To include special characters, enclose in single quotes. Not edited.	These five keywords allow the user to add data not anticipated by the category attributes. Note: Line number is required for DELETE_DATA command.

ELEMENT

GENERAL ATTRIBUTES

No ID keyword required.
Segment name for DELETE_DATA and REPORT commands: DTEATR8

Keyword	Example	Length /Type	Allowable Values	Description
BYTES LENGTH LEN	BYTES=27 LENGTH=300 LEN=12	1-5N	0-32767	Length in bytes of data element as it appears in I/O area. Required for DBD_OUT. Used in STRUCTURES_OUT. DL/I compatible keyword.
TYPE DTETYPE	TYPE=C	1A	B=Bit string C=Alphameric D=Long floating point E=Short floating point F=Binary (4 bytes) H=Binary (2 bytes) P=Packed decimal X=Hexadecimal Z=Zoned decimal	Data element type code. Optional for DBD_OUT. Used for STRUCTURES_OUT. DL/I compatible keyword.
DATE DTEEFDTE	DATE=111279	6N	010100-123199	Effective date of element (Form=MMDDYY). Allows specification of month, day and year in one keyword.
MONTH DTEEFMO	MONTH=12	2N	01-12	Subelement of DATE; effective month.
DAY DTEEFDY	DAY=27	2N	01-31	Subelement of DATE; effective day.
YEAR DTEEFYR	YEAR=75	2N	00-99	Subelement of DATE; effective year.
DTEPREC PRECISION PREC	DTEPREC=(15,8) PRECISION=(31) PREC(20,-1)	1-10 A/N	One or two values 1. First (or only) value can be: 1-255 2. Second value can be: -128 - +127	Specifies the minimum number of significant digits. Optionally includes the position of the binary or decimal point relative to value of a data item. Second value may be substituted for Dictionary keyword DTEPSCL.

Keyword	Example	Length /Type	Allowable Values	Description
DTEPLEN DIGITS PLIPLLEN	DTEPLEN=5 DIGITS=10 PLIPLLEN=255	1-3N	1-255	Minimum number of significant digits.
DTEPSCL DECIMALS PLIPSCS	DTEPSCL=-10 DECIMALS=+21 PLIPSCS=127	1-4A/N	-128 - +127	Position of the binary or decimal point relative to value of a data item. Used by DDT_OUT, RECALCULATE_SEGMENT, and STRUCTURES_OUT.
QSEC MINFQS	QSEC=10 MINFQS=101	1-3A/N	Not edited.	Specifies an access code for access Query security.
USEC MINFUS	USEC=10 MINFUS=101	1-3A/N	Not edited.	Specifies an access code for update access security.

COBOL ATTRIBUTES

No ID keyword required.
Segment name for DELETE_DATA and REPORT commands: DTECATRB

Keyword	Example	Length /Type	Allowable Values	Description
PICTURE PIC	PIC=XX PIC='9(6)'	1-30 A/N	Not edited.	Data for the COBOL PICTURE clause. Used for COBOL STRUCTURES_OUT.
USAGE	USAGE=2	1A/N	0=Computational 1=Computational-1 2=Computational-2 3=Computational-3 4=Computational-4 D=Display I=Index	COBOL USAGE clause data. Used for COBOL STRUCTURES_OUT.
JUST JUSTIFY COBJUST JUSTIFIED	JUST=R	1A	R=Justify right	Data for the COBOL JUSTIFIED clause. Used for COBOL STRUCTURES_OUT.
SYNCH SYNC COBSYNC	SYNCH=SR	1-2A	S=Synchronize SL=Synchronize left SR=Synchronize right	Indicates use of COBOL SYNCHRONIZE clause. Used for COBOL STRUCTURES_OUT.
BLANK BLNK COBBLANK	BLANK=B	1A	B=Blank when zero	Indicates whether data item has COBOL clause BLANK WHEN ZERO. Used for COBOL STRUCTURES_OUT.
SIGN COBSIGN	SIGN=LS	1-2A	L=Sign leading T=Sign trailing LS=Sign leading, separate character. TS=Sign trailing, separate character.	Indicates use of COBOL SIGN clause. Used for COBOL STRUCTURES_OUT.

COBOL VALUE CLAUSE

No ID keyword required.

Segment name for DELETE_DATA and REPORT commands: DTEVALUE

Keyword	Example	Value /Type	Allowable Values	Description
VALUE	VALUE=2753 VALUE='CONST. FIELD' VALUE=SPACES	1-120A/N	1. Any quoted character string 2. Unquoted COBOL keywords: ALL, SPACE, SPACES, ZERO, ZEROES, ZEROS, HIGH-VALUE, HIGH-VALUES, LOW-VALUE, LOW-VALUES, THROUGH, THRU 3. Unquoted numeric values. 4. Multi-word values. Not edited.	Data for the initial value in Assembler or the COBOL VALUE clause. Used for Assembler language and COBOL STRUCTURES_OUT. For correct interpretation by STRUCTURES_OUT, values must conform to rules for COBOL VALUE clauses. Values recorded in Dictionary as entered, including quotes.

COBOL OCCURS CLAUSE

No ID keyword required.

Segment name for DELETE_DATA and REPORT commands: DTELOCC

Keyword	Example	Length /Type	Allowable Values	Description
OCCURS DTELOCC	OCCURS='3 TIMES'	1-120A/N	Not edited.	Data for COBOL OCCURS clause. Used for COBOL, Assembler language, and PL/I STRUCTURES_OUT.

PL/I ATTRIBUTES

No ID keyword required.
Segment name for DELETE_RELATIONSHIP_DATA and REPORT commands: DTEPATR

Keyword	Example	Length /Type	Allowable Values	Description
PLIPIC PPIC	PLIPIC='999'	1-30 A/N	Not edited.	Data for PL/I PICTURE attribute. Used for STRUCTURES_OUT.
PLIINIT INITIAL	PLIINIT=('2','3')	1-30 A/N	Not edited.	Data for PL/I INITIAL attribute. Used for STRUCTURES_OUT.
PLIFMT FMT FORMAT	PLIFMT=BIT PLIFMT=(DEC, FIXED)	3-9A 3-7A	One or two values 1. First (or only) value can be: BINARY BIN N DECIMAL DEC D BIT B CHARACTER CHAR C POINTER PTR P 2. Second value can be: FIXED X FLOAT L V VAR VARYING	Specifies the format of a PL/I arithmetic variable or string variable. Optionally includes the type of variable. Second value may be substituted for Dictionary keyword PLIOPT.
PLIOPT OPT SCALE VAR	PLIOPT=VAR OPT=FLOAT	3-7A	Three values: 1. FIXED X 2. FLOAT L 3. V VAR VARYING	Specifies the type of arithmetic or string variable.
PLIMODE MODE	MODE=R	1-7A	Two values: R REAL C CPLX COMPLEX	Specifies the mode of the arithmetic variable.
PLIALIGN ALIGNED	ALIGNED=Y	1-3A	Two values: Y YES N NO	Specifies whether the element is to be aligned or not.
PLIDIM DIM	DIM=10 DIM=(3,3,3) DIM=(1:10, 1:10,1:5) DIM=(,5:5) PLIDIM= (10,10,-5:5)	1-6A/N 1-43 A/N	Two values: 1. -32767 to 32767. 2. Three dimensions, each can be single value or a pair of values in the range -32767 to 32767.	Specifies the PL/I DIMENSION attribute. 1-3 dimensions can be specified. In each case, a lower and an upper bound can be given. May be substituted for Dictionary keywords DIM1, DIM2, and DIM3.

Keyword	Example	Length /Type	Allowable Values	Description
DIM1 PLIDIM1 DIM2 PLIDIM2 DIM3 PLIDIM3	DIM1=10 DIM2=(1:10) DIM3=(-5:5) PLIDIM1=1 PLIDIM3=(2:2)	1-6A/N 1-15 A/N	Two values: 1. -32767 to 32767 2. A pair of bounds for a dimension, each value in ange -32767 to 32767.	Specifies one of three PL/I DIMENSION attributes. A lower and an upper bound can be given. If only one bound is given, it is assumed to be the upper bound. The lower bound defaults to 1 in this case.
DIM1L PLIDIM1L DIM1U PLIDIM1U DIM2L PLIDIM2L DIM2U PLIDIM2U DIM3L PLIDIM3L DIM3U PLIDIM3U	DIM1L=121 PLIDIM1U=1 PLIDIM2L=0 DIM2U=32767 DIM3L=32767 DIM3U=-10	1-6A/N	-32767 to 32767	Specifies the lower or upper bound of one of three PL/I Dimension attributes. <u>Important</u> : A lower bound may not be specified without the corresponding upper bound.

INQUIRY HEADER AND EDIT DESCRIPTION

ID keyword required: EDITMODE=D, E, or V
Segment name for DELETE_DATA and REPORT commands: DTEEDIT

Keyword	Example	Value /Type	Allowable Values	Description
HDRNO	HDRNO=3	1N	1-9	Specifies the number of header lines.
HDRLEN	HDRLEN=40	1-3N	1-999	Specifies the length of each header line.
HDRTEXT	HDRTEXT='STATUS'	1-120 A/N	Data= free-form text. To include special characters, enclose in single quotes. Not edited.	Specifies the output header for the subject field. HDRLEN defaults to the number of alphanumeric characters in this header.
MASK	MASK='99/99/9'	1-40 A/N	Data= free-form mask. To include the special characters, enclose in single quotes. Not edited.	Specifies an output edit mask for the subject field. Optionally used by DDT_OUT for parameter PATTRN.
EDITMODE EDITSEQ	EDITMODE=V EDITSEQ=3	1A or 1N	Three values: 1. D or 1 for DECODE 2. E or 2 for ENCODE 3. V or 3 for VALIDATE	Specifies the type of edit for the subject field. Either decode before reporting, or encode/validate before storing in a data base.
EDITTYPE TYPSPC	EDITTYPE=E TYPSPC=R	1A	Two values: 1. For edit E,L 2. For validate E,L,R,P	Specifies the type of edit. This can be a user exit (E), a table lookup (L), or, additionally for validate mode, a range check (R) or a mask/picture check (P). TYPSPC is GIS compatible. <u>ID keyword required.</u>
ROUTINE UROUT	ROUTINE=INQEXIT3 UROUT=TBL7	1-8A/N	Not edited.	Specifies the edit routine name for the load module or phase (DOS). UROUT is GIS compatible. <u>ID keyword required.</u>
TABLE EXTABL	TABLE=TBL7 EXTABL=TBL700X	1-8A/N	Not edited.	Specifies the lookup table name for an external module or phase (DOS). EXTABL is GIS compatible. <u>ID keyword required.</u>
EDITUNIT CONVA CONVF	EDITUNIT=C CONVA=Z CONVF=P	1A	C=character string P=packed decimal Z=zoned decimal	Specifies for encode and validate mode (CONVA) the element format. CONVF is for decode. CONVA and CONVF are GIS compatible. <u>ID keyword required.</u>

Keyword	Example	Value /Type	Allowable Values	Description
EDITLEN LGTHA LGTHF	EDITLEN=10 LGTHA=10 LGTHF=8	1-3N	1-255	Specifies for encode and validate mode (LGTHA) the element length in bytes. LGTHF is for decode. LGTHA and LGTHF are GIS compatible. <u>ID keyword required.</u>
EDITUSER EEDVAL	EDITUSER=(5,100) EEDVAL=(5,40)	1N 1-3N	Two values: 1. 1-5 2. 1-999	Specifies the User Data segment and starting line number where edit data is stored for range, picture, or table lookup. ¹ May be substituted for Dictionary keywords EDITNUM and EDITOCC. EEDVAL is GIS compatible. <u>ID keyword required.</u>
EDITNUM	EDITNUM=5	1N	1-5	Specifies the User Data segment for storage of range, picture, or table lookup. <u>ID keyword required.</u>
EDITOCC	EDITOCC=40	1-3N	1-999	Specifies the starting line in a User Data segment for range, picture, or table lookup statements. <u>ID keyword required.</u>
ACTION ERRORD	ACTION=E ERRORD=S	1A	E=exclude field update S=supply zero or blank Not edited.	Specifies a code to indicate action to take if input data fails encode or validate. ERRORD is GIS compatible. <u>ID keyword required.</u>

¹ Edit data statements are stored as text, in strings of 72 characters, with the last occurrence marked by a code word ENDDATA. Arguments are stored in ascending sequence in the form arg. func. arg. func.... or arg. arg.....

LANGUAGE COMMENTARY AND PL/I DATA

ID keyword required: PLISEQ=1-255
 Segment name for DELETE_DATA and REPORT commands: PLIINFO

Keyword	Example	Length /Type	Allowable Values	Description
PLISEQ	PLISEQ=2	1-3N	1-255	Line number of PLIDATA entry. Used for STRUCTURES_OUT.
PLIDATA	PLIDATA='POS(10)' PLIDATA='LAST CHARACTER'	1-72 A/N	To include special characters, enclose in single quotes. Not edited.	72-character comments or narrative text for STRUCTURES-OUT. Also, unsupported attributes for an element. <u>ID keyword required.</u>
PLITYPE	PLITYPE=C	1A	C=Comment N=Narrative X=Unsupported Attribute	Determines meaning of PLIDATA. Used for STRUCTURES_OUT. <u>ID keyword required.</u>

FREE-FORM DESCRIPTION

ID keyword required: TEXTSEQ=1-999

Segment name for DELETE_DATA and REPORT commands: DTEDSCR

Keyword	Example	Length /Type	Allowable Values	Description
DESCRIPTION DESC DTEDSCR	DESC=(1,'SEGMENT DATA BASE') DESC=(2, ' ')	1-3N 1-72 A/N	Two values: 1. n=line number, 1-999 2. Free-form text data. To include special characters, enclose in single quotes. Not edited.	Free-form text describing element.
TEXTSEQ	TEXTSEQ=2	1-3N	1-999	Line number for text. Same as first parameter of DESCRIPTION keyword.
TEXT	TEXT='NAME OF EMPLOYEE'	1-72 A/N	Free-form text. To include special characters, enclose in single quotes. Not edited.	Same as second parameter of DESCRIPTION keyword. <u>ID keyword required.</u>

USER DATA

No ID keyword required.

Segment name for DELETE_DATA and REPORT commands: DTEUSERn

Keyword	Example	Length /Type	Allowable Values	Description
DTEUSER1 DTEUSER2 DTEUSER2 DTEUSER3 DTEUSER4 DTEUSER5	DTEUSER1= (1,'TEXT') DTEUSER3= (1,NOQUOTES) DTEUSER5=(3, ' ')	1-3N 1-80 A/N	Two values: 1. n=line number, 1-999 2. Free-form text data. To include special characters, enclose in single quotes. Not edited.	These five keywords allow the user to add data not anticipated by the Dictionary program. Note: Line number is required for DELETE_DATA command.

PCB**ATTRIBUTES OF A DATA BASE PCB**

No ID keyword required.

Segment name for DELETE_DATA and REPORT commands: PCBATTR

Keyword	Example	Length /Type	Allowable Values	Description
TYPE	TYPE=GSAM	1-4A	D or DB=D G or GSAM=G	TYPE of PCB. Optional for PSB_OUT. DL/I compatible keyword.
KEYLEN	KEYLEN=37	1-4N	1-3825	Number of bytes of longest concatenated key for a hierarchic path of sensitive segments within the logical data structure. Required for PSB_OUT. DL/I compatible keyword.
POS POSITNG	POS=S POSITNG=M	1A	S=Single M=Multiple	Specifies whether single or multiple positioning is desired for accessing logical data structure. Optional for PSB_OUT. DL/I compatible keyword.
PROCSEQ	PROCSEQ=DBSIX2	1-8A/N	Not edited.	Name of secondary index data base used to determine processing sequence in this PCB. Optional for PSB_OUT. DL/I compatible keyword.
PROCOPT	PROCOPT=LS	1-4A	G, I, R, D, A, P, E, L, S, GS, LS, K ² Not edited.	Processing options on sensitive segments declared in this PCB. Required for PSB_OUT. DL/I compatible keyword.
QSEC MINFQS	QSEC=10 MINFQS=101	1-3A/N	Not edited	Specifies an access code for Query access security.
USEC MINFUS	USEC=10 MINFUS=101	1-3A/N	Not edited	Specifies an access code for update access security.

² The letters represent the following values:

G=Get I=Insert R=Replace D=Delete A=All of GIRD
P=Use in GET command code E=Exclusive use
L=Load function (except HIDAM)
S=Segments in ascending sequence only
GS=Get segments in ascending sequence (HSAM only)
LS=Segments loaded in ascending sequence only (HIDAM, HDAM)
K=Key sensitivity only

ATTRIBUTES OF AN OUTPUT MESSAGE PCB

No ID keyword required.

Segment name for DELETE_DATA and REPORT commands: PCBATTRB

Keyword	Example	Length /Type	Allowable Values	Description
TYPE	TYPE=TP	1-4A	T or TP=T	Type of PCB. Optional for PSB_OUT. DL/I-compatible keyword.
EXPRESS	EXPRESS=N	1-3A	Y YES N NO	Specifies whether to send (Yes) or backout (No) a message on program ABEND. Optional for PSB_OUT. DL/I compatible keyword.
MODIFY	MODIFY=N	1-3A	Y YES N NO	Specifies whether TP PCB destination can be modified. Optional for PSB_OUT. DL/I-compatible keyword.
LTERM TRANSMNAME TRNSNAME	LTERM=IMSMSTR TRANSMNAME=MASTER	1-8A/N	Not edited.	Output messages destination; either logical terminal name or transaction code name. Optional for PSB_OUT. DL/I-compatible keyword.
ALTRESP	ALTRESP=YES	1-3A	Y YES N NO	Specifies whether IMS should accept messages to this PCB as conversational responses (IMS/VS 1.1 and higher levels).
SAMETRM	SAMETRM=YES	1-3A	Y YES N NO	Specifies whether IMS/VS should check that this PCB's LTERM is assigned to the same PTERM as the I/O PCB (IMS/VS 1.1 and higher levels).

FREE-FORM DESCRIPTION

ID keyword required: TEXTSEQ=1-999
Segment name for DELETE_DATA and REPORT commands: PCBDSR

Keyword	Example	Length /Type	Allowable Values	Description
DESCRIPTION DESC PCBDSR	DESCRIPTION=(2,'TEXT') DESC=(1,' ')	1-3N 1-72 A/N	Two values: 1. n=line number, 1-999 2. Free-form text data. To include special characters, enclose in single quotes. Not edited.	Free-form text describing PCB.
TEXTSEQ	TEXTSEQ=21	1-3N	1-999	Line number for TEXT. Same as first parameter of DESCRIPTION keyword.
TEXT	TEXT='USED BY PCOS'	1-72 A/N	Free-form text. To include special characters, enclose in single quotes. Not edited.	Same as second parameter of DESCRIPTION keyword. <u>ID keyword required.</u>

USER DATA

No ID keyword required.
Segment name for DELETE_RELATIONSHIP_DATA and REPORT commands: PCBUSERn

Keyword	Example	Length /Type	Allowable Values	Description
PCBUSER1 PCBUSER2 PCBUSER3 PCBUSER4 PCBUSER5	PCBUSER2=(3, '431-2110') PCBUSER4=(12, 'USER DATA') PCBUSER4=(3, ' ')	1-3N 1-80 A/N	Two values: 1. n=line number, 1-999 2. Free-form text data. To include special characters, enclose in single quotes. Not edited.	These five keywords allow user to add data not anticipated by the Dictionary program. Note: Line number is required for DELETE_DATA command.

SEGMENT

GENERAL ATTRIBUTES

No ID keyword required.

Segment name for DELETE_DATA and REPORT commands: SEGATRB

Keyword	Example	Length /Type	Allowable Values	Description
BYTES LENGTH LEN	BYTES=5 LENGTH=(30,20) LEN=3265	1-5N	0-32767	Defines size of fixed-length segment, or maximum and minimum sizes of variable-length segment. May substitute for Dictionary keywords MAXBYTES and MINBYTES. Parens required if two values entered. Required for DBD_OUT. Used for STRUCTURES_OUT. DL/I compatible keyword.
MAXBYTES SEGBYTES	MAXBYTES=34	1-5N	0-32767	Maximum size of variable-length segment. See BYTES keyword. Optional for DBD_OUT.
MINBYTES	MINBYTES=5	1-5N	0-32767	Minimum size of variable-length segment. See BYTES keyword. Optional for DBD_OUT.
NUMELES NUMDTES	NUMELES=7 NUMDTES=10	1-3N	0-255	Number of data elements in segment (user maintained).
DATE SEGEFDTE	DATE=030176	6N	010100-123199	Effective date of segment (MMDDYY).
MONTH SEGEFMO	MONTH=12	2N	01-12	Subelement of DATE; effective month.
DAY SEGEFDY	DAY=03	2N	01-31	Subelement of DATE; effective day.
YEAR SEGEFYR	YEAR=80	2N	00-99	Subelement of DATE; effective year.
ALIGNED SEGALIGN	ALIGNED=Y SEGALIGN=N	1A	Y Yes N No	Specifies whether the segment is aligned or not.

LANGUAGE COMMENTARY AND PL/I DATA

ID keyword required: PLISEQ=1-255
 Segment name for DELETE_DATA and REPORT commands: SGPLINFO

Keyword	Example	Length /Type	Allowable Values	Description
PLISEQ SEGPLIID	PLISEQ=3	1-3N	1-255	Line number of PLIDATA entry. Used by STRUCTURES_OUT.
PLIDATA SGPLDATA	PLIDATA=DIM(4) PLIDATA='LAST CHARACTER'	1-72 A/N	To include special characters, enclose in single quotes. Not edited.	72-character card images of output for STRUCTURES-OUT function, or narrative describing output. Do not include DECLARE verbs, level numbers or punctuation. <u>ID keyword required.</u>
PLITYPE SGPLTYPE	PLITYPE=N	1A	C=Comment N=Narrative X=Unsupported attribute	Meaning of PLIDATA entry. Used by STRUCTURES_OUT. <u>ID keyword required.</u>

FIELDS USED FOR SECONDARY INDEX

No ID keyword required.
Segment name for DELETE_DATA and REPORT commands: XFLD1

Keyword	Example	Length /Type	Allowable Values	Description
XFLDNAME	XFLDNAME=XFLD1	1-8A/N	Not edited.	XFLD name, used in SSAs when accessing indexed segment in indexed data base. Optional for DBD_OUT. DL/I compatible keyword.
SEGMENT ISSNAME	SEGMENT=SEG1 ISSNAME=SEG2	1-8A/N	Not edited.	Name of Index Source Segment containing all fields used in creating secondary index. Optional for DBD_OUT. DL/I-compatible keyword.
NULLVAL	NULLVAL=X'21' NULLVAL=C'9' NULLVAL=BLANK	1-5A/N	C' '=any valid character except C'B' or C'Z' X'nn'=any valid hex value except X'C2' or X'E9' B or any string beginning with B= blank Z or any string beginning with Z= zero	Specifies conditions under which an indexed entry is suppressed. See <u>IMS/VS Utilities Reference Manual</u> . Optional for DBD_OUT. DL/I-compatible keyword.
EXTRTN	EXTRTN=INDXRTN	1-8A/N	Not edited.	Name of routine to process inserts, deletes, or replacements to index data base which results from changes to indexed data base. Optional for DBD_OUT. DL/I compatible keyword.
CONST XCONST	CONST=C'3' CONST=X'00'	1A/N	C' '=any valid character except blanks X'nn'=any valid hex value (not X'40').	Constant which precedes each index entry key. Used for shared index feature. Optional for DBD_OUT. DL/I compatible keyword.

SRCH FIELDS FOR GENERATING KEY FIELD OF SECONDARY INDEX DATA BASE ROOT SEGMENT

ID keyword required: SRCHOCR=1-5
Segment name for DELETE_DATA and REPORT commands: XFLD2

Keyword	Example	Length /Type	Allowable Values	Description
SRCHOCR	SRCHOCR=5	1N	1-5	Order of occurrences of 'SRCH' fields in XDFLD statements. Optional for DBD_OUT.
SRCHNAME	SRCHNAME=DTE3 SRCHNAME=(P,A,DTE3,0)	1-8A/N 1A/N 1A 1-8A/N 1-3N	User name, or (Status, Subject code, User name, Occurrence) See Figure 2.	13-byte subject name of data element of Index Source Segment regarded as source data for indexed data base. Used in generating key field of root segment of secondary index data base. Default qualifiers entered if no parens. Optional for DBD_OUT. <u>ID keyword required.</u>

SUBSEQ AND DDATA FIELDS FROM INDEX SOURCE SEGMENT

ID keyword required: SBSROCR=1-5³
Segment name for DELETE_DATA and REPORT commands: XFLD3³

Keyword	Example	Length /Type	Allowable Values	Description
SBSROCR	SBSROCR=3	1N	1-5	Order of occurrences of SUBSEQ and DDATA fields in XDFLD statement. Optional for DBD_OUT.
DDATA	DDATA=(P,A, DTE1,0) DDATA=DTE1	1-8A/N 1A/N 1A 1-8A/N 1-3N	User name, or (Status, Subject code, User name, Occurrence) See Figure 2.	13-byte subject name of DDATA element in index source segment. Don't use in same command with SUBSEQ. Default qualifiers entered if no parens. Optional for DBD_OUT. <u>ID keyword required.</u>
SUBSEQ	SUBSEQ= (P,A,DTE,0) SUBSEQ=DTE4	1-8A/N 1A/N 1A 1-8A/N 1-3N	User name, or (Status, Subject code, User name, Occurrence) See Figure 2.	13-byte subject name of SUBSEQ element in Index Source Segment. Don't use in same command with DDATA. Default qualifiers entered if no parens. Optional for DBD_OUT.

³ When using the DELETE_DATA command, you must specify two id values. The first value must be 'R' when deleting DDATA and 'Q' when deleting SUBSEQ data. The second value must be a valid value for the SBSROCR keyword. For example, DATA=((XFLD3,Q,3)) refers to the third segment of SUBSEQ data.

FREE-FORM DESCRIPTION

ID keyword required: TEXTSEQ=1-999
Segment name for DELETE_DATA and REPORT commands: SEGDSR

Keyword	Example	Length /Type	Allowable Values	Description
DESC DESCRIPTION SEGDSR	DESC=(1, ' ') DESCRIPTION= ('3, PROGRAMMER NAME')	1-3N 1-72 A/N	Two values: 1. n=line number, 1-999 2. Free-form text data. To include special characters, enclose in single quotes. Not edited.	Free-form text describing segment.
TEXTSEQ	TEXTSEQ=5	1-3N	1-999	Line number for TEXT. Same as first parameter of DESCRIPTION keyword.
TEXT	TEXT='UNUSUAL SEGMENT'	1-72 A/N	Free-form text. To include special characters, include in single quotes. Not edited.	Same as second parameter of DESCRIPTION keyword. <u>ID keyword required.</u>

USER DATA

No ID keyword required.
Segment name for DELETE_DATA and REPORT commands: SEGUSERn

Keyword	Example	Length /Type	Allowable Values	Description
SEGUSER1 SEGUSER2 SEGUSER3 SEGUSER4 SEGUSER5	SEGUSER2=(21, 'COMPACTION LOGIC') SEGUSER4= (5,'THE SOURCE DOCUMENT') SEGUSER4=(3, ' ')	1-3N 1-80 A/N	Two values: 1. n=line number, 1-999 2. Free-form text data. To include special characters, enclose in single quotes. Not edited.	These five keywords are provided to allow user to add data not anticipated by the Dictionary program. Note: Line number is required for the DELETE_DATA command.

SYSTEM

ATTRIBUTES OF THE PROGRAM

No ID keyword required.
Segment name for DELETE_DATA and REPORT commands: SYSATRB

Keyword	Example	Length /Type	Allowable Values	Description
PGMLANG PGMLNG	PGMLANG=P PGMLNG=C	1A	A through Z. Reserved values: A=Assembler C=COBOL P=PL/I F=FORTRAN	Indicates language in which program is written.
PGMSIZE SIZE	PGMSIZE=10000 SIZE=2710	1-8N	0-16777215	Size of program.
PGMTYPE TYPE	PGMTYPE=D TYPE=T	1A	T=TP B=BMP D=DB or Batch	Program type.

ATTRIBUTES OF THE PSB

No ID keyword required.
Segment name for DELETE_DATA and REPORT commands: SYSATRB

Keyword	Example	Length /Type	Allowable Values	Description
LANG PSBLANG PSBLNG	LANG=P PSBLANG=A	1-3A	A=Assembler C=COBOL P=PL/I	Compiler language in which application program using PSB written. Required for PSB_OUT. DL/I compatible keyword.
MAXQ PSBMAXQ	MAXQ=29 PSBMAXQ=40	1-5N	0-32767	Number of DBS calls with QX status codes which may be issued between SYNCH points. Optional for PSB_OUT. DL/I-compatible keyword.
IOASIZE PSBIOSZ	IOASIZE=1000 PSBIOSZ=5000	1-5N	0-32767	Largest I/O area to be used by application program. Optional for PSB_OUT. DL/I compatible keyword.
CMPAT PSBCMPAT	CMPAT=Y PSBCMPAT=N	1-3 A	Y YES N NO	Indicates whether I/O PCB is always present. Optional for PSB_OUT. DL/I compatible keyword.
IOEROPN	IOEROPN=451 IOEROPN=(18, WTOR)	1-4N 4A	One or two values 1. 0-4095 2. WTOR	First value specifies condition code to be returned if any I/O errors on data bases (or 451 for ABEND) WTOR specifies operator message. (IMS/V5 1.1 or higher level, batch only.)
IOWTR	IOWTR=W IOWTR='	1A/N	W or blank	Same as second value of IOEROPN. Specifies operator message if any I/O errors on data bases. (IMS/V5 1.1 or higher level, batch only.)
SSASIZE PSBSSASZ	SSASIZE=107	1-5N	0-32767	Maximum total length of all SSAs to be used by the application program. Optional for PSB_OUT. DL/I compatible keyword.
PSBTYPE	PSBTYPE=BATCH	1-5A	B BATCH T TIP	Describes the type of application program being defined. Optionally used for STAGE_1_OUT.
OLIC	OLIC=Y	1-3A	Y Yes N No	Specifies whether PSB is to allow online image copy or not. Optional for PSB_OUT.

Keyword	Example	Length /Type	Allowable Values	Description
OVLY	OVLY=YES	1-3A	Y YES N NO	Indicates if the application program uses overlay design. Optionally used for STAGE_1_OUT.
IQF	IQF=YES	1-3A	Y YES N NO	Specifies if the program invoked is IQF (Interactive Query Facility). Optional for STAGE_1_OUT.

ATTRIBUTES OF THE TRANSACTION

No ID keyword required.
Segment name for DELETE_DATA and REPORT commands: SYSATRB

Keyword	Example	Length /Type	Allowable Values	Description
IMSTTYPE TRTYPE	IMSTTYPE=T TRTYPE=L	1A	L=LTERM T=TRAN CODE	Type of transaction.
LTERMTYPE LTERMTYP LTTYPER	LTERMTYPE=M LTTYPER=M	1A	M=Master terminal	Type of logical terminal.
WFI	WFI=NO	1-3A	Y YES N NO	Indicates if this is a wait-for-input transaction. Optionally used for STAGE_1_OUT.
MSGSEG	MSGSEG=MULT	1-4A	M MULT S SNGL	Indicates if the incoming message can be more than one segment long. Optional for STAGE_1_OUT.
MSGRESP	MSGRESP=YES	1-3A	Y YES N NO	Indicates if the communication line from which the transaction was entered is to be held until a response is received. Optional for STAGE_1_OUT.
INQTRAN	INQTRAN=NO	1-3A	Y YES N NO	Specifies whether or not this is an inquiry transaction. Optionally used for STAGE_1_OUT.
INQRECV	INQRECV=YES	1-3A	Y YES N NO	Indicates if this inquiry transaction should be recovered during an IMS/VSE restart (valid only if INQTRAN=YES). Optional for STAGE_1_OUT.
MODE	MODE=SNGL	1-4A	M MULT S SNGL	Indicates condition upon which data base buffers are to be flushed. Optional for STAGE_1_OUT.
EDITNAME	EDITNAME=MSGEDIT	1-8A/N	Not edited	Name of a user-supplied transaction input-edit routine which edits messages prior to being received by the program. Optionally used for STAGE_1_OUT.

Keyword	Example	Length /Type	Allowable Values	Description
EDITTRAN	EDITTRAN=ULC	1-3A	U UC L ULC	Defines transaction as entered from terminal as being uppercase/lowercase or translatable to uppercase prior to presentation to processing program. Optional for STAGE_1_OUT.
SEGNO	SEGNO=10	1-5N	0-65535	Specifies maximum number of application program output segments allowed into message queues per GU call from the application program. Optionally used for STAGE_1_OUT.
SEGSIZE	SEGSIZE=1024	1-5N	0-65535	Specifies maximum number of bytes allowed in each output segment. Optional for STAGE_1_OUT.
SPASIZE	SPASIZE=768	1-5N	0-65535	Specifies size of the conversational scratch pad area. Optionally used for STAGE_1_OUT.
SPAFIX	SPAFIX=YES	1-3A	Y YES N NO	Indicates if same size SPA used by all transactions during course of this conversation. Optional for STAGE_1_OUT.

FREE-FORM DESCRIPTION

ID keyword required: TEXTSEQ=1-999
Segment name for DELETE_DATA and REPORT commands: SYSDESCR

Keyword	Example	Length /Type	Allowable Values	Description
DESCRIPTION DESC SYSDESCR	DESC= (3,'EMPLOYEE DATA') DESCRIPTION= (5,'XXXXXXX')	1-3N 1-80 A/N	Two values: 1. n=line number, 1-999 2. Free-form text data. To include special characters, enclose in single quotes. Not edited.	Free-form text describing subject.
TEXTSEQ	TEXTSEQ=7	1-3N	1-999	Line number for TEXT. Same as first parameter of DESCRIPTION keyword.
TEXT	TEXT='DESCRIPTIVE DATA'	1-72 A/N	Free-form text. To include special characters, enclose in single quotes. Not edited.	Same as second parameter of DESCRIPTION keyword. <u>ID keyword required.</u>

USER DATA

No ID keyword required.
Segment name for DELETE_DATA and REPORT commands: SYSUSERn

Keyword	Example	Length /Type	Allowable Values	Description
SYSUSER1 SYSUSER2 SYSUSER3 SYSUSER4 SYSUSER5	SYSUSER1= (1,'PROGRAM FLOW') SYSUSER3=(17, 'JOB RUN INFO.')	1-3N 1-80 A/N	Two values: 1. n=line number, 1-999 2. Free-form text data. To include special characters, enclose in single quotes. Not edited.	These five keywords allow user to add data not anticipated by the Dictionary program. Note: Line number is required for the DELETE_DATA command.

APPENDIX B: DICTIONARY KEYWORDS FOR RELATIONSHIP DATA

This appendix defines all the keywords used to specify relationship data for pairs of standard categories. These keywords are used with the ADD_RELATIONSHIP command and the CHANGE_RELATIONSHIP_DATA command. The keywords are presented in a series of tables. Each table and its continuation, names all the keywords appropriate for a particular pair of standard subject categories. The major headings in the table are the types of relationship for which relationship data can be entered: for example, SYSDEF-DATABASE, or PSB-TRANSACTION. Under the major headings, the keywords are grouped by the segments in which the relationship data is stored in the Dictionary data base.

One or more segments, containing the data associated with the group of keywords, can be deleted by a DELETE_RELATIONSHIP_DATA command. The segment names are given under each table's subheading. Some keywords are associated with the basic relationship entry and the data can be deleted only with the DELETE_RELATIONSHIP command, which deletes the specified relationship and all relationship data associated with the relationship.

The segment names are also used to specify a group of data in a report. One or more segment names can be used for the DETAIL= operand in the REPORT command.

Users of display forms referring to these tables should understand the different requirements for subject names and differences in free-form text entries in the display forms.

Notes:

1. The first column gives the keyword name and alternative names; the second column gives one or more examples of the keyword's use; and the last three columns give the length and type of entry, allowable values, and a description of the meaning and use of the keyword.
2. In the Length and Type column, A=alphabetic, N=numeric, and A/N=alphameric.
3. An entry in the Description column noting that a keyword is DL/I-compatible always refers to the first keyword in column 1.
4. Segment names marked with an asterisk (*) can be deleted only with the DELETE_RELATIONSHIP command.
5. "Not edited" means that only length and type are validated.
6. Required ID keywords are listed at the top of the page.

DATABASE WITH SEGMENT

GENERAL CHARACTERISTICS

No ID keyword required.
Segment name for REPORT command: DBSSEG/SEGDBS*

Keyword	Example	Length /Type	Allowable Values	Description
PARENT	PARENT=SEG1 PARENT=(P,A,SEG1,0) PARENT=((SEG1,SNGL)) PARENT=((SG1),(SEG2,VIRTUAL,DBS2)) PARENT=0		See example. Subject name or same as in DBDGEN. User name restricted to 8 A/N characters.	Parent of this segment in this data base. Logical and physical may be specified: No parens: physical parent, user name only. 1 paren: subject name of physical parent. 2 parens: One to three values: 1.Physical parent, user name 2.PCPNTRS (optional) 3.LPARENT (optional) PARENT=0 implies LEVEL=1. Optional for DBD_OUT. DL/I compatible keyword.
PCPNTRS	PCPNTRS=D	1A	S=Single D=Double ' '=neither single nor double	Indicates whether segment has single or double physical child pointers from parent. Optional for DBD_OUT. Same as PARENT=((name, SNGL or DBLE)).
PHYSPARENT PHYPAR PPNAME	PHYSPARENT=(P,A,SEG1,0) PHYPAR=SEG1	1-8A/N 1A/N 1A 1-8A/N 1-3N	User name, or (Status, Subject code, User name, Occurrence) See Figure 2.	13-byte subject name of physical parent of this segment. Default qualifiers inserted if no parens. (See PARENT keyword.) This keyword, or PARENT required for DBD_OUT.
DSG SEGDSGNO	DSG=1 SEGDSGNO=1	1-2N	1-10	Dictionary number of data set group to which this segment belongs. Required for DBD_OUT.
LEVEL SEGLVL SEGLEVEL	LEVEL=1 SEGLVL=10 SEGLEVEL=3	1-2N	1-15	Level number of segment within this data base. LEVEL=1 implies PARENT=0.

Keyword	Example	Length /Type	Allowable Values	Description
TYPE SEGTYPE	TYPE=P SEGTYPE=V	1A	V=Virtual P=Physical L=Logical X=Primary Index Pointer Y=Secondary Index Pointer X and Y must match data base.	Describes segment type. Defines segment role in data base. Optional for DBD_OUT.
PAIREDLC	PAIREDLC=P	1A	P=Paired ' '=Not paired	Indicates whether segment is paired logical child. Optional for DBD_OUT. Same as PTR=PAIRED, which is stored in DBSSEG.

PHYSICAL ATTRIBUTES OF SEGMENT

No ID keyword required.
 Segment name for DELETE_RELATIONSHIP_DATA and REPORT commands:
 SEGPATRB

Keyword	Example	Length /Type	Allowable Values	Description
FREQ SEGFREQ	SEGFREQ=30000 FREQ=.02 FREQ=2.3	1-8N	.10-16777215	Count of segments of this type expected to occur within a parent. Optional for DBD_OUT. DL/I compatible keyword.
POINTER PTR	POINTER=(T,LP,LTB) PTR=(NOTWIN) PTR=(HIERBWD, PAIRED)		1. [HP] HIER or H HIERBWD or HB 2. [PTP] TWIN or T TWINBWD or TB NOTWIN or NT 3. [LTP] LTWIN or LT LTWINBWD or LTB 4. [LPP] LPARNT or LP 5. [CTR] CTR 6. [PAIREDLC] PAIRED	Pointer options of segment specified. Optional for DBD_OUT. DL/I-compatible keyword. (PTR=PAIRED is actually stored in DBSSEG segment.) Each numbered group is equivalent to the bracketed keyword which appears above it. A single pointer can be specified or multiple pointers can be specified in a parenthesized list. For a specification of multiple pointers, only one value can be chosen for each group.
CTR	CTR=C CTR=' '	1A	C=CTR is to be reserved in segment prefix ' '=No CTR to be reserved	Indicates whether a CTR is to be reserved in the segment prefix, as in POINTER=CTR. Optional for DBD_OUT.
HIERPNTR HP HPNTRS	HIERPNTR=B HP=H	1A	H=hierarchic B=hierarchic backward and forward	Indicates whether segment has hierarchic pointers, and of what type. Optional for DBD_OUT. Same as PTR=H, or PTR=HB.
LTWINPNTR LTP LTPNTRS	LTWINPNTR=L LTP=B	1A	L=logical twin forward B=logical twin forward and backward	Indicates whether segment has logical twin pointer, and of what type. Optional for DBD_OUT. Same as PTR=LT, or PTR=LTB.
LPARPNT LPP LPARNT	LPARPNT=P LPP=' '	1A	P=logical parent pointer present ' '=pointer not present	Indicates whether segment has a logical parent. Same as POINTER=LP. Optional for DBD_OUT.

Keyword	Example	Length /Type	Allowable Values	Description
PTWINPNTR PTP TPNTRS	PTWINPNTR=T PTP=B	1A	T=physical twin B=physical twin backward and forward N=no physical twin pointers	Indicates whether segment has physical twin pointers, and of what type. Same as PTR=T, or PTR=TB, or PTR=NT. Optional for DBD_OUT.
RULES	RULES=(PLP,HERE) RULES=(,LAST) RULES=PBP		One or two values, same as DL/I: 1.Insert (P,L,V), delete (P,L,V,B), and replace (P,L,V) codes written as three-character string. 2.Location code FIRST, LAST, HERE or any string beginning with F, L, or H.	Specifies insert, delete, replace, and location rules for segment in this DBS. Specified as in DBDGEN. May substitute for Dictionary keywords IRULE, DRULE, RRULE, WHRRULE. Optional for DBD_OUT. DL/I compatible keyword.
IRULE INSRULE	IRULE=P INSRULE=V	1A	P=Physical L=Logical V=Virtual	Insert rule for segment within this data base. Same as first character of RULES value. Optional for DBD_OUT.
DRULE DELRULE	DRULE=B DELRULE=P	1A	P=Physical L=Logical V=Virtual B=Bidirectional	Delete rule for segment within this data base. Same as second character of RULES value. Optional for DBD_OUT.
RRULE REPRULE	RRULE=V REPRULE=P	1A	P=Physical L=Logical V=Virtual	Replace rule for segment within this data base. Same as third character of RULES value. Optional for DBD_OUT.
WHRRULE	WHRRULE=F	1A	F=First L=Last H=Here	Defines where new occurrences of segment are to be inserted. Same as second value of RULES keyword. Optional for DBD_OUT.
COMPRTN	COMPRTN=RTN1 COMPRTN=(RTN, KEY, INIT) COMPRTN=(RTN, DATA)	1-8A/N 1-4A 4A	One to three values, same as DL/I: 1. Routine name 2. (CCRTNOPT) KEY or K, DATA or D 3. COMPINIT value. See example or appropriate DL/I Utilities Reference Manual. Not edited.	Name of compression routine used when storing this segment. Values for CCRTNOPT and COMPINIT may be specified if parentheses used, as in DBDGEN. Optional for DBD_OUT. DL/I compatible keyword.

Keyword	Example	Length /Type	Allowable Values	Description
CCRTOPT	CCRTOPT=K	1A	K=Key D=Data	Compression routine options of key or data. Same as second parameter of COMPRTN. Optional for DBD_OUT.
COMPINIT	COMPINIT=I	1A	I=generate an INIT for DBD_OUT	INIT option of compression routine. Same as third parameter of COMPRTN. Optional for DBD_OUT.

PHYSICAL CHILD SEGMENT

ID keyword required: PCSGNAME
Segment name for DELETE_RELATIONSHIP_DATA and REPORT commands:
SEGPCHLD

Keyword	Example	Length /Type	Allowable Values	Description
SEGPCHLD	SEGPCHLD=(P, A,SEG4,0)	1-8A/N	User name, or	13-byte subject name of the physical child. Default qualifiers inserted if no parens.
PCSGNAME	PCSGNAME=SEG3	1A/N	(Status,	
		1A	Subject code,	
		1-8A/N	User name,	
		1-3N	Occurrence) See Figure 2.	

LOGICAL CHILD SEGMENT

ID keyword required: LCSGNAME
Segment name for DELETE_RELATIONSHIP_DATA and REPORT commands:
SEGLCHIL

Keyword	Example	Length /Type	Allowable Values	Description
LCHILD	LCHILD=(SEG1, DBS1) LCHILD=SEG1	1-8A/N 1-8A/N	One or two values 1. User name of logical child segment (required). 2. User name of data base (optional). Names restricted to eight alphameric characters. Default qualifiers inserted. Same as LCSGNAME and LCDBNAME.	Logical child segment and data base (optional). Default qualifiers are inserted. Includes LCSGNAME and LCDBNAME and may be substituted for either in AR and CRD commands. Optional for DBD_OUT. DL/I-compatible keyword.
LCSGNAME	LCSGNAME=(P, A,SEG1,0) LCSGNAME=SEG1	1-8A/N 1A/N 1A 1-8A/N 1-3N	User name, or (Status, Subject code, User name, Occurrence) See Figure 2.	13-byte subject name of logical child segment. Default qualifiers inserted if no parens. LCHILD is alternate form. Optional for DBD_OUT.
LCDBNAME	LCDBNAME=(P, P,DBS1,0) LCDBNAME=DBS1	1-8A/N 1A/N 1A 1-8A/N 1-3N	User name, or (Status, Subject code, User name, Occurrence) See Figure 2.	13-byte subject name of data base in which logical child segment contained. Default qualifiers inserted if no parens. Optional for DBD_OUT. <u>ID keyword required.</u>
POINTER PTR	POINTER=INDX POINTER=SYMB PTR=D	1-4A	S SNGL D DBLE I INDX Y SYMB	Pointer options for logical parent segments to this logical child segment. Relates to logical child. Meaning differs from other POINTER keyword. Optional for DBD_OUT. DL/I-compatible keyword. <u>ID keyword required.</u>

Keyword	Example	Length /Type	Allowable Values	Description
INDEX INDXFLNM	INDEX=EMPNO INDXFLNM=SSAN	1-8A/N	Not edited.	Name of field being indexed if this logical segment is used to define a primary index relationship. If secondary, contains XDFLB field name. This parameter appears as part of the LCHILD specification (for the index target segment) associated with the relationship between the index pointer segment and the index data base. Optional for DBD_OUT. <u>ID keyword required.</u>
SECINDX	SECINDX		Enter keyword and ID or omit.	Indicates whether this logical child defines a secondary index relationship. This parameter appears as a part of the LCHILD specification (for the secondary index pointer segment) associated with the relationship between the segment that is being indexed (the target segment) and its data base. Optional for DBD_OUT. <u>ID keyword required.</u>
PAIR PAIRNAME	PAIR=(P,A,SEG1,0) PAIRNAME=SEG1	1-8A/N 1A/N 1A 1-8A/N 1-3N	User name, or (Status, Subject code, User name, Occurrence) See Figure 2.	13-byte subject name of segment with which logical child segment is paired. Default qualifiers inserted if no parens. Optional for DBD_OUT. <u>ID keyword required.</u>
LCIRULES	LCIRULES=L	1A	F=First L=Last H=Here	Rule for inserting new occurrences of logical child segment. Optional for DBD_OUT. <u>ID keyword required.</u>

LOGICAL PARENT SEGMENT

No ID keyword required.
 Segment name for DELETE_RELATIONSHIP_DATA and REPORT commands:
 SEGLPATR

Keyword	Example	Length /Type	Allowable Values	Description
LPARENT	LPARENT=(SEG1, PHYSICAL,DBS1) LPARENT=(SEG1, V,DBS1)	1-8A/N 1A 1-8A/N	One to three values. 1.User name of logical parent segment (required) 2.Key storage option (optional) P=Physical V=Virtual 3.User name of data base.	Logical parent segment, data base, and key storage option. Data base must be specified. Same as LPSGNAME, KEYSTRG, and LPDBNAME. See PARENT keyword in DBSSEG segment. Optional for DBD_OUT. DL/I compatible keyword.
LPSGNAME	LPSGNAME=(P,A, SEG1,0) LPSGNAME=SEG2	1-8A/N 1A/N 1A 1-8A/N 1-3N	User name, or (Status, Subject code, User name, Occurrence) See Figure 2.	13-byte subject name of logical parent segment. Default qualifiers inserted if no parens. Optional for DBD_OUT.
KEYSTRG	KEYSTRG=P	1A	P=Physical V=Virtual	Indicates whether key of logical parent is to be physically stored as part of segment. Optional for DBD_OUT.
LPDBNAME	LPDBNAME=(P,P, DBS1,0) LPDBNAME=DBS1	1-8A/N 1A/N 1A 1-8A/N 1-3N	User name, or (Status, Subject code, User name, Occurrence) See Figure 2.	13-byte subject name of data base to which logical parent belongs. Default qualifiers inserted if no parens. Optional for DBD_OUT.

SOURCE SEGMENT (VIRTUAL OR LOGICAL SEGMENT)

ID keyword required: SRCOCR=1-2
Segment name for DELETE_RELATIONSHIP_DATA and REPORT commands:
SEGSRC

Keyword	Example	Length /Type	Allowable Values	Description
SOURCE1	SOURCE1=(SG1, KEY,DBS1)	1-8A/N	1.User name	First subparameter of SOURCE= keyword. May substitute for dictionary keywords SRCOCR=1, SRCSNAME, SRCKDOPT, and SRCNAME. Default qualifiers filled into subject name. Source base must be entered. See SOURCE2. Optional for DBD_OUT.
	SOURCE1=(SEG, D,DBS1)	1-4A	2.KEY or K, DATA or D	
	SOURCE1=(SEG1,,DBS1)	1-8A/N	3.User name of segment	
SOURCE2	SOURCE2=(SEG1, K,DBS1)	1-8A/N	1.User name of segment	Second subparameter of SOURCE= keyword. Like SOURCE1, but SRCOCR=2. May not be specified in same command as SOURCE1, or other equivalent keywords. Optional for DBD_OUT.
	SOURCE2=(SEG1, DATA,DBS1)	1-4A	2.KEY or K, DATA or D (optional)	
	SOURCE2=(SG1,,DBS1)	1-8A/N	3.User name of data base (optional)	
SRCOCR	SRCOCR=2	1N	1=first 2=second	Specifies whether first or second source argument. Same data entered in SOURCE1, SOURCE2. Optional for DBD_OUT.
SRCSNAME	SRCSNAME=SEG1 SRCSNAME=(P,A, SEG1,0)	1-8A/N	User name, or	13-byte subject name of segment which is SOURCE for this segment. Default qualifiers inserted if no parens. See SOURCE1, SOURCE2. Optional for DBD_OUT. <u>ID keyword required.</u>
		1A/N	(Status,	
		1A	Subject code,	
		1-8A/N	User name,	
		1-3N	Occurrence) See Figure 2.	
SRCKDOPT	SRCKDOPT=K	1A	K=key only D=key and data	Indicates whether key, or key and data portion of the second source argument are to be used in constructing this segment. Same as second value of SOURCE1 or SOURCE2. Optional for DBD_OUT. <u>ID keyword required.</u>

Keyword	Example	Length /Type	Allowable Values	Description
SRCDNAME	SRCDNAME=(P,P, DBS1,0) SRCDNAME=DBS1	1-8A/N	User name, or	13-byte subject name of a data base in which source segment is contained. Default qualifiers inserted if no parens. See SOURCE1 and SOURCE2. Optional for DBD_OUT. <u>ID keyword required.</u>
		1A/N	(Status,	
		1A	Subject code,	
		1-8A/N	User name,	
		1-3N	Occurrence)	
			See Figure 2.	

PCB WITH SEGMENT

GENERAL CHARACTERISTICS

No ID keyword required.
Segment name for REPORT commands: PCBSEG/SEGPCB*

Keyword	Example	Length /Type	Allowable Values	Description
PARENT	PARENT=(P,A, SEGI,0) PARENT=SEG PARENT=0	1-8A/N 1A/N 1A 1-8A/N 1-3N	User name, or (Status, Subject code, User name, Occurrence) See Figure 2.	13-byte subject name of the parent of this segment in logical hierarchy. Default qualifiers inserted if no parens. Parent=0 generates (P, A, 0, 0); equivalent to parent not specified. Required for DBD_OUT. DL/I compatible keyword.
PROCOPT SEGPROPT	PROCOPT=GIRD PROCOPT=LS	1-4A	G=Get I=Insert R=Replace D=Delete A=All the above P=Command code D will be used on Get calls E=Exclusive use (for G, I, R, D, A) L=Load (except HIDAM) K=Key sensitivity only S=Ascending sequence only (used with G, I, D, R, A, L) GS=Get in ascending sequence only (HSAM only) LS=Loaded in ascending sequence only (HIDAM, HDAM) K=Key sensitive only Not edited.	Processing options allowed for PCB using this sensitive segment. May contain up to four values from list. Optional for PSB_OUT. DL/I-compatible keyword.

INDEX USED IN SSAS

ID keyword required: INDCOCR=1-32
Segment name for DELETE_RELATIONSHIP_DATA and REPORT commands:
SNSGINDC

Keyword	Example	Length /Type	Allowable Values	Description
INDEX	INDEX=(1,DBS1) INDEX=(21,DBSIND2)	2N 1-8A/N	Two values: 1. 1-32 2. Name of secondary index data base	Specifies INDCOCR and INDCNAME in one keyword. Represents one entry of PCB INDICES list. Optional for PSB_OUT. DL/I compatible Keyword.
INDCOCR	INDCOCR=10	1-2N	1-32	Indicates order of an INDCNAME entry within list of indexes. Same as first value of INDEX keyword. Optional for PSB_OUT.
INDCNAME	INDCNAME=DBS1	1-8A/N	Not edited.	Name of secondary index data base where XDFLDS may be used in SSAs for the SENSEG in this PCB. Same as second value of INDEX keyword. One entry of PCB INDICES list. Optional for PSB_OUT. <u>ID keyword required.</u>

SEGMENT WITH ELEMENT

GENERAL CHARACTERISTICS

ID keyword required: START=1-32767
Segment name for REPORT command: SEGDTE/DTESEG*

Keyword	Example	Length /Type	Allowable Values	Description
START	START=1 START=255	1-5N	1-32767	Starting position of data element in this segment. REQUIRED IN EVERY COMMAND PERTAINING TO THIS RELATIONSHIP. Used in structures only.
BITSTART BST DTEBISTR	BITSTART=7 BST=3 DTEBISTR=2	1N	1-8	Starting bit position in the byte specified by the START value of the element in the segment. <u>ID keyword required.</u>
FIRSTLEV	FIRSTLEV=A	1A/N	Not edited.	Reserved, may be considered User Data. <u>ID keyword required.</u>
SEQIND SEQ GENFLD	SEQIND=A SEQ=M GENFLD=' '	1A	U=Sequence field with unique values only M=Sequence field allowing duplicate values G=Not a sequence field, but generate a FIELD statement A=Ascending sequence non-DL/I D=Descending sequence non-DL/I 6=COBOL level 66 ' '=Undefined	Indicates whether element is a sequence field in this segment. Also indicates non-sequence fields which appear in DBDGEN. A value of U, M, or G will cause a FIELD statement to be generated by the DBD_OUT command. <u>ID keyword required.</u>

ELEMENT CONTAINS ELEMENT

GENERAL CHARACTERISTICS

ID keyword required: START=0-32767

Segment name for REPORT command: DTECNTNS/DTECNTNW*

Keyword	Example	Length /Type	Allowable Values	Description
START SUBSTART	START=3 SUBSTART=21	1-5N	0-32767 for USE=R or 8 1-32767 otherwise	Start position of this element with another element. REQUIRED IN EVERY COMMAND REFERRING TO AN ELEMENT CONTAINS ELEMENT RELATIONSHIP.
BITSTART BST BITSTRT	BITSTART=4 BST=1 BITSTRT=3	1N	1-8	Starting bit position in the byte specified by the SUBSTART value of the element within the containing segment. <u>ID keyword required.</u>
USE DTESUID	USE=D DTESUID=A	1A/N	Any alphabetic character, but the following have special meaning: C=CONTAINS D=DEPENDS R=REDEFINES Also valid: 6=RENAMES 8=condition names (COBOL level 88)	Type of relationship. Any alphabetic character can be defined by installation. C, R, 6, and 8 used by STRUCTURES_OUT. For USE=8, START is automatically set to zero. <u>ID keyword required.</u>

SYSDEF WITH DATABASE

GENERAL CHARACTERISTICS

No ID keyword required.
Segment name for REPORT command: SYSDBS/DBSSYS*

Keyword	Example	Length /Type	Allowable Values	Description
DMBRES	DMBRES=YES	1-3A	Y YES N NO	Indicates if DMB should be made resident at system initialization time. A value of Y will cause positional parameter RESIDENT to be produced on DATABASE macro statement by STAGE_1_OUT.

SYSDEF WITH PSB**GENERAL CHARACTERISTICS**

No ID keyword required.
Segment name for REPORT command: SYSSYS*

Keyword	Example	Length /Type	Allowable Values	Description
PSBLOAD	PSBLOAD=RES	1-4A	R RES D DOPT	Specifies if PSB associated with this application program to be made resident at system initialization time. Optional and used by STAGE_1_OUT.
CLASS	CLASS=14	1-3N	1-255	Defines the class to which following transaction codes are to be assigned. Optional and used by STAGE_1_OUT.
SCHDTYP	SCHDTYP=SERIAL	1-8A	S SERIAL P PARALLEL	Specifies if this application program can be scheduled into more than one message or BMP region simultaneously. Optional, used by STAGE_1_OUT.
RSYSID	RSYSID=1	1-3N	1-255	Specifies the remote system identification of the system in a multiple IMS/VS system configuration. Optional, used by STAGE_1_OUT.
LSYSID	LSYSID=255	1-3N	1-255	Specifies the local system identification of the system in a multiple IMS/VS system configuration. Optional, used by STAGE_1_OUT.

SYSDEF WITH TRANSACTION

GENERAL CHARACTERISTICS

No ID keyword required.
Segment name for REPORT command: SYSSYS*

Keyword	Example	Length /Type	Allowable Values	Description
PRTYNORM	PRTYNORM=1	1-2N	0-14	Defines transaction's priority when count of enqueued transactions is less than the limit count value. Optional, used by STAGE_1_OUT.
PRTYLIMT	PRTYLIMT=14	1-2N	0-14	Defines priority to which this transaction is raised when enqueued transaction count equal or greater than the limit count value. Optional, used by STAGE_1_OUT.
LIMTCONT	LIMTCONT=1	1-5N	0-65535	Defines number which determines if normal or limit priority value assigned to this transaction. Optional, used by STAGE_1_OUT.
PROCNUM	PROCNUM=200	1-5N	0-65535	Defines the number of messages of this transaction code a program can process in a single scheduling. Optional, used by STAGE_1_OUT.
PROCSECS	PROCSECS=512	1-5N	0-65535	Defines the amount of time (in seconds) allowable to process a single transaction. Optional, used by STAGE_1_OUT.
PARLIM	PARLIM	1-5N	0-32767	Specifies the threshold value to be used to determine when to schedule an additional message processing region. Optional, used by STAGE_1_OUT.
MSGCLASS	MSGCLASS=7	1-3N	1-255	Specifies the class to which this transaction code is assigned. Optional, used by STAGE_1_OUT.
SCHD	SCHD=1	1N	1-4	Specifies scheduling option used for other transactions when this one is not schedulable. Optional, used by STAGE_1_OUT.

Keyword	Example	Length /Type	Allowable Values	Description
SPAKEEP	SPAKEEP=CORE	1-4A	C CORE D DASD	Specifies if SPA is to be kept in CORE or on direct access device. Optional, used by STAGE_1_OUT.

PSB WITH TRANSACTION

GENERAL CHARACTERISTICS

No ID keyword required.
Segment name for REPORT command: SYSSYS*

Keyword	Example	Length /Type	Allowable Values	Description
OSTTYPE	OSTTYPE=I	1-6A	I INPUT O OUTPUT	Specifies the type of transaction defined with this PSB.

DICTIONARY MESSAGES AND CODES

This appendix lists all the Dictionary system messages, in message ID order, and abnormal termination codes. With each message there is further explanation of the cause of the message, if necessary, plus a brief description of the system action, if any, and the suggested user response, if required.

DICTIONARY ABNORMAL TERMINATION CODES

During Dictionary operation it is possible to encounter a serious error in the Dictionary system itself. The only way to obtain sufficient information to locate and solve the problem is to get a full abnormal termination dump.

Additionally, you may have occasion to want a dump as an aid in solving a problem noted by a Dictionary produced message. The three message tables:

DBDMTBLX - common messages
DBDMTBLO - online messages
DBDMTBLB - batch messages

include an abnormal termination request flag. You can code the MSGSKL macro for any message to request an abnormal termination by specifying DUMP=Y (see the DB/DC Data Dictionary Diagnosis: Reference). When the message is issued, an abnormal termination is taken. The abnormal termination request flag is independent of the severity level of the message.

The following abnormal termination codes are issued by the Dictionary:

1000 OPEN/CLOSE REQUEST CODE INVALID

Explanation: The output routine is entered for only three reasons, (1) to open a file, (2) to close a file, and (3) to prepare for output. The code in the request area indicates none of these. Register 3 points to the HDRDSECT in which this request area is found.

1500 DL/I CHANGE CALL FAILED

Explanation: The Change call, which was issued to set the destination of an output operation, completed with a nonblank status code. Register 4 points to the PCB in which the status code may be found. This code will only be issued in an online system.

2000 UNABLE TO OPEN DCB

Explanation: The output routine was unsuccessful in an attempt to open a file. Register 6 points to the DCB for the file. Examine the DCBOFLGS for reason for the failure. This code will only be issued in a batch system.

2500 DL/I INSERT CALL FAILED

Explanation: The Insert call, which was issued to build an output message, completed with a nonblank status code. Register 4 points to the PCB in which the status code may be found. This code will only be issued by an online system.

3000 INVALID DESTINATION CODE

Explanation: An invalid destination code was specified for an open, close, or output operation. Register 3 points to the HDRDSECT in which the invalid code may be found.

3500 ATTEMPT TO SEND OUTPUT WITH ZERO OR NEGATIVE LENGTH

Explanation: In a request to send a line of output, the subtraction of the length of the LLZZ area from the length of the intended output line resulted in a negative or zero length. Register 5 contains the calculated message length. Register 3 points to the trailer area containing the input length and the message to be sent.

3800 DBDWGMAT RECEIVED BAD REQUEST CODE

Explanation: DBDWGMAT expects one of a subset of the allowable retrieval request codes to be set in the PACA "PATYPE" field. Normally this would be guaranteed by module DBDWVCA. The problem may be caused by an error in DBDWGMAT itself or DBDWVCA, unless some other module has chosen to call DBDWGMAT directly.

User Response: Notify system support personnel.

3802 LINKAGE MODULE RECEIVED BAD PROGRAM ACCESS entry pointer

Explanation: One of the Program Access Language Linkage Modules, DBDWLNKA, DBDWLNKP, or DBDWLNKC, received a zero address value for the entry point to the Program Access routines. One possible reason for this would be that a routine which was not expected to issue calls to Program Access was invoked via the linkage module and did, in fact, attempt to call Program Access.

User Response: Notify system support personnel.

3804 DBDWPAK RECEIVED BAD REQUEST TYPE

Explanation: PATYPE value is neither a retrieve nor a write request. This should have been caught by DBDWVCA, unless it was bypassed.

User Response: Notify system support personnel.

3806 DBDWIFA RECEIVED UNEXPECTED RETURN CODE FROM PROGRAM ACCESS

Explanation: Either the calling program passed bad input in the PACA or other arguments, or an error occurred in Program Access operation. Register 15 at time of abend contains the PACA return code.

User Response: Notify system support personnel.

3808 DBDWRA FAILED TO FIND SEGMENT IN DBDSEGS TABLE

Explanation: An internal Dictionary problem has occurred.

User Response: Notify your IBM representative.

4000 REQUESTED ABEND

Explanation: Either the user has requested an abnormal termination by setting the abnormal termination request flag in a message definition or the Dictionary message function encountered an error from which it cannot recover and has requested an abnormal termination. This abnormal termination is always preceded by an attempt to output a message, either DBD2304 for errors encountered in the message function or any other message that has been coded to request abnormal termination.

DICTIONARY NUMBERED MESSAGES

During Dictionary operation you receive printed or displayed messages, some simply informational, and others requiring some action on your part. Each message is preceded by a message identification:

DBDmmnn t
DBD

the component code that distinguishes Dictionary messages from DL/I system messages

mm

identifies the Dictionary function that caused the message

nn

a message sequence number within each function

t

identifies the type of message, as follows:

- I** Information message. (Return code=0)
- W** Warning. Execution continues. (Return code=4)
- E** Error. Function not executed. Error can be corrected by user. (Return code=8)
- S** Severe error during execution. Other commands may be executed. (Return code=12)
- T** Termination error. This is a system error that prevents execution of further commands. (Return code=16)

You may receive more than one message in a single run or online operation. You receive just one condition code, which corresponds to the most serious type code in the messages issued.

Note: The type code is not applicable when the formatted display facility is used. In addition, only one message, the one corresponding to the most serious type code, is displayed to the interactive display forms facility user.

GENERAL SYSTEM MESSAGES (DBD00NN T)

DBD0001 I PROCESSING CONTINUES

Explanation: An error has been detected in the processing of the current command and is described in the message preceding this message.

System Action: Processing continues for this command.

User Response: Check preceding messages for information.

DBD0002 E name dbd/psb CANNOT BE ADDED

Explanation: A serious error has occurred during the processing of the current DBD_IN or PSB_IN command. The error is described in the message preceding this message.

System Action: Execution ends for this command. Any subsequent commands are processed.

User Response: Check preceding messages for information.

DBD0003 I name dbd/psb NOW IN
DICTIONARY

Explanation: Processing for the current DBD_IN or PSB_IN command has completed successfully and all applicable entities and their relationships and attributes have been entered in the Dictionary.

System Action: Execution ends for this command.

User Response: None.

DBD0004 W name SEGMENT HAS NO 'DL/I'
ALIAS

Explanation: This segment is a Dictionary primary name without a DL/I alias.

System Action: A segment statement is produced with an invalid name, and execution continues.

User Response: Repair the DBDGEN source produced. Use the Update commands to provide the segment with the proper alias.

DBD0005 W POSSIBLE ERRORS IN
function-name OUTPUT

Explanation: A serious error (for example, a required attribute such as field length is missing) has occurred during the processing of the current DBD_OUT or PSB_OUT command.

System Action: Execution ends for this command. Any subsequent commands are processed.

User Response: Check preceding messages for information.

DBD0006 T UNDEFINED RETURN CODE FROM
PROC-name

Explanation: An invalid return code was detected by one of the language translator execution routines.

System Action: Execution of the command is terminated.

User Response: This problem should not occur. Notify your IBM representative.

DBD0007 T OPEN ERROR - name

Explanation: An error occurred when opening an output file or library file.

System Action: Dictionary processing is terminated

User Response: Check the specification on the DD card for this output file and correct all errors. Then resubmit the job.

DBD0008 S INVALID SUBJECT CODE

Explanation: An invalid subject category was entered.

User Response: Enter the command with the correct subject category.

DBD0009 E subject category subject
name DOES NOT EXIST

Explanation: The subject name could not be found.

System Action: Execution is terminated.

User Response: Verify that the subject category and subject name were correctly specified, or use the ADD command to enter subject name and data.

DBD0010 I THE PRIMARY NAME FOR
subject name IS subject
name

Explanation: This message gives the primary name when another name is entered in the command. Even though many names may have the same meaning, all data is actually stored as dependent segments of the primary name.

System Action: Execution continues.

User Response: None.

DBD0011 E RELATIONSHIP DOES NOT
EXIST

Explanation: There is no relationship between the two subject names in the command.

System Action: Execution of the command is terminated. No change is made to Dictionary data bases.

User Response: Verify that the subject categories and subject names were entered correctly, or use the ADD_RELATIONSHIP command to establish the relationship.

DBD0012 I RELATIONSHIP DOES NOT
EXIST

Explanation: The relationship could not be deleted, because no relationship could be found for the subject names and starting position specified.

User Response: Verify that the subject names and starting position were correctly specified.

DBD0013 T UNEXPECTED STATUS CODE cc
AFTER action IN MODULE
module name

Explanation: Program malfunction. The module received an unexpected status code from a DL/I call.

System Action: Execution is terminated.

User Response: Check that all PSBs and DBDs for the Dictionary are available to the system. Check the invoking JCL for errors. Check that the Dictionary data bases have been created properly. If it is an online system, ensure that the system was properly brought up. If no problem can be found, notify your IBM representative regarding this error.

DBD0014 S INPUT COMMAND TO MODULE
module name IS NOT E OR R

Explanation: An invalid command was entered.

System Action: Execution is terminated.

User Response: Verify the correct spelling of the command, then repeat the command.

DBD0015 I subject category subject
name IS name

Explanation: The only purpose of this message is to establish a reference name.

System Action: Execution continues.

User Response: None.

DBD0016 W field-name WAS NOT CHANGED
BECAUSE 'FROM' VALUE DOES
NOT MATCH DICTIONARY VALUE

Explanation: This field-name is not stored in the Dictionary as it was specified in the FROM portion of the command.

System Action: The field is not changed. Execution continues with the next field in input.

User Response: Verify that field-name was specified as desired or use the EXISTING option of the CHANGE_IN command.

DBD0017 W field-name WAS NEVER ADDED

Explanation: The segment containing this field name is not stored in the Dictionary.

System Action: Execution continues. For subjects in standard categories, no data is added. For subjects in installation-defined categories, the field value will be added to the Dictionary.

User Response: Use the ADD command to add a value for field-name.

DBD0018 I RELATIONSHIP WAS DELETED

Explanation: The command was executed as requested.

System Action: This message indicates that execution of the command has terminated successfully.

User Response: None.

DBD0019 I RELATIONSHIP ESTABLISHED
UNDER PRIMARY NAMES

Explanation: The command was executed as requested. The relationship is actually between primary names even if alias names were used in the command.

System Action: Execution of the command has terminated successfully. Relationship was added.

User Response: None.

DBD0020 E FUNCTION NOT PERFORMED,
RELATIONSHIP ALREADY
EXISTS

Explanation: The alias relationship between the two input subjects has already been established in the Dictionary.

System Action: Execution of the command is terminated. No change is made to Dictionary contents.

User Response: Verify that the subject categories and subject names were correctly specified.

DBD0021 W field-name HAS ALREADY
BEEN ADDED

Explanation: This field-name is already stored in the Dictionary.

System Action: Execution continues with the next field in input. No change is made to existing data.

User Response: Use the CHANGE_RELATIONSHIP_DATA command to alter the value for field-name if this is an ADD_RELATIONSHIP command, or use the CHANGE_IN command to alter the value for field-name if this is an ADD command. If this message appears when you are working on the segment fields form of the interactive display forms facility, it means that you have added a field to the list with a MODE of 0 (old), and have supplied values for type and/or bytes. However,

information is already stored in the Dictionary for at least one of these parameters. In this case, you must use blanks to delete the data on the screen for the offending parameter before selecting the next action. When the screen has been completely processed the first time, the stored values for type and bytes will be displayed. If this message appears during batch forms execution, it indicates that you have specified an attribute for a subject with a MODE of 0 which already has data stored for it in the Dictionary. Values cannot be changed using the batch forms, so the commands mentioned above should be used if a change is desired.

DBD0022 E segment-name IS AN UNDEFINED OPERAND

Explanation: This segment-name is not a valid Dictionary segment for the subject category specified.

System Action: Execution of the command is terminated.

User Response: Verify that the subject name, segment name, and ID (if required), were specified correctly.

DBD0023 W DATA name DOES NOT EXIST

Explanation: This segment-name could not be found in the Dictionary.

System Action: Execution of the command is terminated.

User Response: Verify that the name was specified correctly.

DBD0024 W NO DATA WAS DELETED

Explanation: No subject data or relationship data has been deleted from the Dictionary.

System Action: Execution of the command has been terminated. No change has been made to Dictionary data.

User Response: Check prior messages and verify that the DATA keyword was included.

DBD0025 W NO DATA WAS ADDED

Explanation: No subject data or relationship data has been added to the Dictionary.

System Action: Execution of the command has terminated.

User Response: Check prior messages for additional information.

DBD0026 I OK**

Explanation: A card was found to contain only Dictionary processing control keywords, and no command verb.

All global functions were performed as specified.

System Action: Processing continues with the next command.

DBD0027 W NO DATA WAS CHANGED

Explanation: Either the user did not specify data to be changed, or prior messages indicate problems with the keywords that were specified.

System Action: Processing continues.

User Response: Correct the problems indicated by the prior messages.

DBD0028 I RELATIONSHIP ALREADY EXISTS

Explanation: The user used an ADD_RELATIONSHIP command to add a relationship or data to a relationship which has already been stored in the Dictionary.

System Action: The relationship was not added.

User Response: None, if this was the desired result. Otherwise, verify that the command was entered properly.

DBD0029 E ID field name FOR SEGMENT segment name WAS EITHER NOT SPECIFIED OR SPECIFIED AS NULL

Explanation: An ID is required when fields in this segment are being added.

System Action: Execution continues with the next command.

User Response: Enter command including ID for this segment with an appropriate value.

DBD0030 I ID field name FOR SEGMENT segment name WAS EITHER NOT SPECIFIED OR SPECIFIED AS NULL IN THE TO PORTION, SO THE FROM VALUE WAS USED

Explanation: The assumption was made that the user did not intend to change the ID.

System Action: Execution continues.

User Response: If this action was not as desired, enter the command and the ID with appropriate values in both the FROM and TO portions of the command.

DBD0031 I ID field name FOR SEGMENT segment name WAS EITHER NOT SPECIFIED OR SPECIFIED AS NULL IN THE FROM PORTION, SO THE TO VALUE WAS USED.

Explanation: The assumption was made that the user did not intend to change the ID.

System Action: Execution continues.

User Response: If this action was not as desired, enter the command and the ID with appropriate values in both the FROM and TO portions of the command.

DBD0032 E ID field name FOR SEGMENT segment name WAS EITHER NOT SPECIFIED OR SPECIFIED AS NULL IN BOTH THE FROM AND TO PORTIONS OF THE COMMAND

Explanation: An ID is required in the FROM and/or TO portions of the command when fields in this segment are being changed.

System Action: Execution continues with the next command.

User Response: Enter the command and the ID with appropriate values in the FROM and/or TO portions of the command.

DBD0033 I ID field name FOR SEGMENT segment name WAS EITHER NOT SPECIFIED OR SPECIFIED AS NULL
- SO A VALUE OF 0 WILL BE USED

Explanation: The ID is used to locate the proper relationship so an ID value of 0 will be used.

System Action: Execution continues.

User Response: If the relationship is not found, the ID should be included to locate the proper relationship.

DBD0034 W DICTIONARY DOES NOT STORE OR OUTPUT THE DEVADDR PARAMETER OF HSAM TAPE DBDS

Explanation: The parameter DEVADDR can be specified in DL/I DOS/VS DBDs with ACCESS=HSAM and DEVICE=TAPE but the Dictionary does not maintain this information. This message is issued by DBD_IN and DBD_OUT to warn of this condition.

System Action: Execution continues.

User Response: No response is necessary for DBD_IN, but you should be aware of the condition. For DBD_OUT you will have to manually modify the DBDGEN to add the desired DEVADDR parameter.

DBD0035 I Add-name WAS NEVER ADDED

Explanation: The value which was found in the Dictionary for the specified keyword is null. This means that the

value was never added.

System Action: The value specified in the 'to' portion of the command will be added.

User Response: None.

DBD0040 S SYSTEM ERROR -- INVALID FUNCTION PASSED TO MODULE name

Explanation: The internal code passed to the module named is invalid input to that module. If the module specified is DBDIBAOL, the function was not Get Unique (GU) or Get Next (GN). If DBDLPROC, an invalid execution function was specified.

System Action: Execution is terminated.

User Response: This message signals an internal Dictionary problem. Notify your IBM representative.

DBD0041 T SYSTEM ERROR code IN MODULE module name

Explanation: An unrecoverable system error has occurred in the indicated module (module name) at the point indicated by "code." The purpose of forcing this abnormal termination is to provide a dump from which the error can be diagnosed.

System Action: After this message is issued, an abnormal termination is forced.

User Response: The abnormal termination dump should be shown to your IBM representative.

DBD0042 S SYSTEM ERROR. RETURN CODE rc WAS RETURNED TO MODULE calling-mod FROM MODULE called-mod

Explanation: The return code which was returned to an internal Dictionary module indicates that an error occurred because of an unrecoverable Dictionary or DL/I error. If it is caused by a DL/I error, an additional message will be issued to indicate the nature of that error.

System Action: Execution of the command is terminated.

User Response: If the return code indicates a Dictionary error, notify your IBM Representative. If this is a DL/I error, correct the data base in error and resubmit the command.

DBD0060 W A COMMAND SEGMENT HAD A NEGATIVE LENGTH

Explanation: Module DBDGPPC has discovered that a command has been built incorrectly.

System Action: An indication of the error is returned to the calling module and processing continues.

User Response: This message indicates a serious internal Dictionary problem. Consult your IBM representative.

**DBD0061 W A COMMAND GREATER THAN
 number BYTES WAS RECEIVED**

Explanation: Module DBDGPPC has been passed a command which is too big for its processing.

System Action: An indication of the error is returned to the calling module and processing continues.

User Response: This message indicates a serious internal Dictionary problem. Consult your IBM representative.

**DBD0065 W COULD NOT FIND DATABASE
 name IN THE PCB LIST**

Explanation: Module DBDGNFO could not find the data base name that was passed to it in the Dictionary-formatted PCB list.

System Action: A return code is set and control is returned to the calling module.

User Response: This message indicates a serious internal Dictionary problem. Consult your IBM representative.

**DBD0066 W AN ERROR STATUS CODE WAS
 ENCOUNTERED ON A 'GO' CALL
 TO THE name DATABASE.
 STATUS CODE: code**

Explanation: Module DBDGNFO has received a DL/I error return code while attempting to find a free occurrence number for a Dictionary subject.

System Action: A return code is set, and control is given to the calling module.

User Response: This message indicates a serious internal Dictionary or DL/I problem. Consult your IBM representative.

**DBD0090 E SUBJECT name DOES NOT
 EXIST**

Explanation: This subject could not be found.

System Action: No relationship was established.

User Response: Use the ADD command to enter subject name into the Dictionary; then, repeat the command.

LANGUAGE TRANSLATOR (DBD0INN T)

**DBD0101 S UNEXPECTED STATUS (cc)
 FROM DL/I**

Explanation: An unexpected status code was returned by DL/I while attempting to read Dictionary input.

System Action: Execution is terminated.

User Response: This message may indicate a problem with the Dictionary's PSB. If the PSB is correct, notify your IBM representative, as some internal Dictionary problem may exist.

**DBD0102 E UNDEFINED OPERAND -
 operand**

Explanation: An invalid keyword or keyword value was detected.

System Action: Statement execution is terminated.

User Response: Check spelling of the keyword for the system. Correct the keyword, and resubmit the job.

**DBD0103 E KEYWORD OR OPERAND OMITTED
 - keyword/operand**

Explanation: The command does not contain the specified keyword or operand.

System Action: Execution of the command is terminated.

User Response: Correct the command and resubmit the job.

**DBD0104 E DELIMITER TO END VALUE NOT
 FOUND - keyword/operand**

Explanation: One of the following errors has been detected:

- A comma or right parenthesis was not found to terminate a value in a list of values for a keyword
- The list of values specified for a keyword is too long
- The end of the command was reached before an operand was found
- A semicolon (;) appeared within a quoted string, but the command was not terminated with a semicolon.

The error was detected following the portion of the command shown in the message.

System Action: Command execution is terminated.

User Response: Correct the command and resubmit the job.

**DBD0105 S SYSTEM ERROR, PROCESSING
TERMINATED DUE TO INVALID
VFD TABLE**

Explanation: A processing table has an invalid value field description.

System Action: Execution is terminated. If tables have not been altered.

User Response: Make sure that the Dictionary tables have not been altered incorrectly. If they have not, notify your IBM representative.

DBD0106 E OPERAND - value - TOO LONG

Explanation: The operand named has exceeded the maximum valid length specified by the Dictionary. This message may also be issued if the keyword operand should be a fully qualified name, but only a user-name was specified.

System Action: Execution of the command is terminated.

User Response: Consult the DB/DC Data Dictionary Administration and Customization Guide for valid operand values. For Extensibility attributes, consult the appropriate GUIDE report. Correct the operand value and resubmit.

**DBD0107 E OPERAND - name - NOT
NUMERIC**

Explanation: The value specified in the message is not numeric and a numeric value was anticipated by the dictionary.

System Action: Execution is terminated.

User Response: Correct the operand, and resubmit the job.

**DBD0108 E OPERAND - name - BELOW
RANGE**

Explanation: The operand is below the allowable range.

System Action: Execution is terminated.

User Response: Correct the operand, and resubmit the job.

**DBD0109 E OPERAND - name - ABOVE
RANGE**

Explanation: The operand is above the allowable range.

System Action: Execution is terminated.

User Response: Correct the operand, and resubmit the job.

**DBD0110 E OPERAND - name - NOT VALID
NAME**

Explanation: The keyword operand is not a valid name.

System Action: Execution is terminated.

User Response: Correct the operand and resubmit the job.

**DBD0111 E OPERAND - name - NOT VALID
CODE**

Explanation: One of the following errors may cause this message to be issued:

- The keyword specified is invalid
- The value of the keyword operand does not match any of the valid values documented.
- The status code or subject code in a fully qualified subject-name is invalid.

System Action: Execution is terminated.

User Response: Correct the operand, and resubmit the job.

**DBD0112 I UNABLE TO COMPLETE
REQUESTED FUNCTION**

Explanation: The requested function was not performed because of previous errors in the command.

System Action: Syntax is checked on the following commands if FLUSH=YES has been indicated.

User Response: Correct all errors, and resubmit the job.

**DBD0113 S REQUESTED FUNCTION MODULE
NOT IN SYSTEM**

Explanation: The requested processing module was not included at linkage edit time.

System Action: Processing is terminated.

User Response: Re-link edit the Dictionary system load module including the desired processing module.

**DBD0114 T UNRECOVERABLE PROCESSOR
ERROR**

Explanation: An error occurred in one of the processing modules.

System Action: Processing is terminated.

User Response: Correct all errors, and resubmit the job. If failure recurs notify your IBM representative.

DBD0115 S ERROR OCCURRED DURING OUTPUT PROCESSING

Explanation: An error was detected in the output processor.

System Action: Processing is terminated.

User Response: Correct all errors, and resubmit the job.

DBD0116 I FUNCTION NOT PERFORMED, SYNTAX CHECK ONLY

Explanation: This function was not completely performed as specified, although it may have been partially performed. If there was an error in the previous command, the current command was only checked for proper syntax.

User Response: Correct the command in error and resubmit those commands not performed.

DBD0117 E INVALID USE OF COMMENT CARD

Explanation: A comment card has been found within a command. A comment card may not appear between continuation cards of a command.

System Action: The command is not executed.

User Response: Place the comment card before or after the command cards.

DBD0118 S SPA SIZE INCORRECT

Explanation: The SPA size specified is not the size the Dictionary requires.

System Action: Application is terminated.

User Response: Find the needed SPA size in the installation manual, and correct it on the transaction definition.

DBD0120 I VALLNGTH NO LONGER USED

Explanation: VALLNGTH is no longer a valid keyword. The length of the data item is computed from the VALUE clause.

System Action: Execution continues.

User Response: If the VALUE keyword was specified, the command was correct.

DBD0121 E INVALID FOR keyword PARAMETER

Explanation: An invalid value was detected for the specified keyword.

System Action: Execution of the command is terminated.

User Response: Change the value in error, and resubmit the command.

DBD0122 I value DUPLICATED FOR keyword PARAMETER, IGNORED

Explanation: The same value has been detected more than once for the specified keyword.

System Action: The duplicate value is ignored, and execution continues.

User Response: None

DBD0125 E EXPECTED CONTINUATION NOT RECEIVED

Explanation: The previous card contains a + which indicates that a continuation card is to follow. The expected continuation card was not received.

System Action: The command is not executed.

User Response: Either insert the continuation card or remove the + sign.

DBD0131 W MORE THAN 50 KEYWORDS ENTERED. NO FURTHER CHECKING FOR DUPLICATES

Explanation: The table which is used in checking for duplicates is filled.

System Action: Execution continues.

User Response: If you habitually enter more than 50 keywords in a command, call your IBM representative for details on enlarging the table.

DBD0132 W KEYWORD ENTERED MORE THAN ONCE - name -

Explanation: A previously entered keyword or its synonym has been entered.

System Action: The new value is used.

User Response: If the value just entered is correct, no action is necessary. Otherwise, correct it.

DBD0133 E RELTYPE HAS NOT BEEN INSTALLED

Explanation: The Dictionary cannot find an installed definition for the relationship specified between categoryA and categoryB.

System Action: Execution of the command is terminated.

User Response: The user should specify a different relationship type or install a relationship type between categoryA and categoryB with the specified keyword.

**DBD0134 E SPECIFIED ALIAS
RELATIONSHIP IS INVALID**

Explanation: There are two conditions which may cause this message to be issued:

- The subject categories for the two input subjects are not the same. This is an error because an alias relationship can only be specified between subjects in the same subject category.
- The alias relationship involves System objects. This is an error because System objects cannot have aliases.

System Action: Execution of the command is terminated.

User Response: The user should correct the command and resubmit it.

**DBD0135 E SEQUENCE ATTRIBUTE FOR
RELATIONSHIP WAS NOT
SPECIFIED**

Explanation: The relationship specified in the command is a sequenced relationship. This means that it must be identified by a sequence attribute value.

System Action: Execution of the command is terminated.

User Response: Consult the appropriate Guide report to determine the name of the sequence attribute and its valid values. Include the sequence attribute value in the command, and resubmit the command.

**DBD0136 I SEQUENCE ATTRIBUTE FOR THE
RELATIONSHIP WAS NOT
SPECIFIED IN THE FROM
PORTION OF THE COMMAND**

Explanation: The relationship specified in the command is a sequenced relationship. This means that it must be identified by a sequence attribute value. The sequence attribute value was not specified in the 'FROM' portion of a change command.

System Action: The value specified in the 'TO' portion of the command will be used.

User Response: No action required.

**DBD0137 I SEQUENCE ATTRIBUTE FOR THE
RELATIONSHIP WAS NOT
SPECIFIED IN THE TO
PORTION OF THE COMMAND**

Explanation: The relationship specified in the command is a sequenced relationship. This means that it must be identified by a sequence attribute value. The sequence attribute value was not specified in the 'TO' portion of a change command.

System Action: The value specified in the 'FROM' portion of the command will be used.

User Response: No action required.

**DBD0138 E LENGTH OF OPERAND value
LESS THAN MINIMUM VALID
LENGTH**

Explanation: The length of the operand value is less than the minimum valid length defined for that value. Valid operand lengths have been defined for the subject names and the attributes of Extensibility subjects.

System Action: Execution of the command is terminated.

User Response: Consult the appropriate Guide report for valid operand values. Correct the operand value, and resubmit the commands.

**DBD0139 E DATA TYPE OF OPERAND value
INVALID**

Explanation: The data type of the operand does not match the data type expected by the Dictionary.

System Action: Execution of the command is terminated.

User Response: Consult the appropriate GUIDE report for valid attribute values. Correct operand value and resubmit.

**DBD0140 E INSTANCE NUMBER value IS
INVALID**

Explanation: The instance number specified for a repeating attribute was outside the valid range of values. This value must be greater than zero, and less than or equal to the maximum number of times this attribute may be repeated.

System Action: Execution of the command is terminated.

User Response: Specify a numeric value between 1 and the REPEAT value of the attribute, inclusively. Resubmit the command.

**DBD0141 E KEYWORD keyword IS
INCORRECT**

Explanation: The format of the keyword has been specified incorrectly. This can be caused by the following conditions:

- The keyword does not begin with an alphabetic character or contains a character which is not alphameric,
- An equal sign (=) does not follow the keyword, or
- The instance number identifying a repeating attribute has been omitted or has been incorrectly specified.

System Action: Execution of the command is terminated.

User Response: Consult Dictionary GUIDE reports to determine the proper format for attribute keywords.

DBD0142 E INSTANCE NOT SPECIFIED FOR KEYWORD keyword

Explanation: The attribute keyword being processed has been defined as a repeating attribute. This means that the attribute must always be identified by an instance number.

System Action: Execution of the command is terminated.

User Response: The instance number enclosed in parentheses should be specified immediately after the keyword. Correct the keyword specification, and resubmit the command.

DBD0143 E KEYWORD keyword IS NOT A REPEATING ATTRIBUTE

Explanation: An instance number was specified for an attribute which does not repeat.

System Action: Execution of the command is terminated.

User Response: Consult Dictionary Guide reports to determine the proper format for the keyword in error. Correct the command, and resubmit the command.

DBD0144 S REQUESTED USER PROGRAM userprogram NOT FOUND

Explanation: An EXECUTE command has specified a user program which is not in the library or a required validation routine is not in the library.

System Action: A severe error condition return code is set. This may result in flushing the command stream.

User Response: Be sure that the user program name is spelled correctly and that the JCL specifies the correct

library.

DBD0145 S ERROR CODE code FROM SYSTEM MACRO macro IN MODULE module

Explanation: An error occurred during execution of the identified macro.

System Action: A severe error condition return code is set. This may result in flushing the command stream.

User Response: Check the documentation of the specified system macro for details of the indicated return code.

DBD0146 S INSUFFICIENT SPACE TO LOAD USER PROGRAM userpgm

Explanation: There was not enough space left in the Dictionary region to bring in the requested program.

System Action: A severe error condition return code is set. This may result in flushing the command stream.

User Response: Increase the Dictionary region size.

DBD0147 S userpgm IS NOT A VALID PROGRAM FOR USE WITH THE DICTIONARY

Explanation: The requested load module or phase must meet the requirements set forth in chapter 10, "Writing Programs to Access Dictionary Data" in the DB/DC Data Dictionary Administration and Customization Guide. Most probable cause is that the required linkage module is not the entry point of the load module, or the program name is misspelled in the command.

System Action: A severe error condition return code is set. This may result in flushing the command stream.

User Response: Ensure that the required linkage module is the primary entry point of the requested program, and that the input command was correct.

DBD0150 S INVALID RETURN CODE code FROM USER PROGRAM userpgm

Explanation: The return code from the user-written program was not one of the allowable values: 0, 4, 8, 12, 16; or the return code from a validation routine is not 0, 4, or 8.

System Action: A severe error condition return code is set. This may result in flushing the command stream.

User Response: Notify the user's programming staff.

DBD0151 I user-provided message

Explanation: This message was provided by a user-written routine, and was issued as a result of validating a subject name or an attribute value.

System Action: Execution continues.

User Response: Consult documentation on output from user-written routines.

DEB0152 W user-provided message

Explanation: This message was provided by a user-written routine, and was issued as a result of validating a subject name or an attribute value.

System Action: Execution continues.

User Response: Consult documentation on output from user-written routines.

DEB0153 E user-provided message

Explanation: This message was provided by a user-written routine, and was issued as a result of validating a subject name or an attribute value.

System Action: Execution of the command is terminated.

User Response: Consult documentation on output from user-written routines.

DEB0156 I ERROR FOUND WHEN VALIDATING ATTRIBUTE keyword

Explanation: This message is issued when a validation routine for an attribute is not found or when the validation type for an installation-defined attribute is invalid. It identifies which attribute was currently being processed.

System Action: Execution of the command is terminated.

User Response: Check preceding messages for information.

DEB0157 S VALIDATION TYPE INVALID

Explanation: The validation type which is stored in the Dictionary for an attribute is invalid. This value may only be 'LIST', 'RNG', and 'RTN'.

System Action: Execution of the command is terminated.

User Response: Make sure no installation changes have been made to modify Dictionary contents. If no changes have been made, notify your IBM representative.

DEB0158 I INSTANCE NUMBER FOR ATTRIBUTE IS number.

Explanation: This message identifies the instance of the repeating attribute specified in the previous

message.

System Action: None.

User Response: None.

DEB0170 E RELATIONSHIP WOULD CAUSE LOSS OF STRUCTURAL INTEGRITY

Explanation: The relationship was not established because of a conflict concerning the use of the status code.

System Action: The command was not executed. Subsequent commands are flushed with a syntax check only, unless FLUSH=NO.

User Response: Verify that subject names were correctly specified. Use COPY to change the status code.

DEB0171 E CHANGE_NAME NOT VALID FOR STATUS CHANGE - USE COPY

Explanation: CHANGE_NAME must not be used to change the status code of any subject name because it does not preserve structural integrity.

System Action: The command was not executed. Subsequent commands are flushed with a syntax check only, unless FLUSH=NO.

User Response: Enter the COPY command to change the status code.

DEB0172 E ALIAS RELATIONSHIP ONLY VALID FOR SAME STATUS

Explanation: An alias relationship can only be established between subject names with the same status. Therefore the relationship requested was not established.

System Action: The command was not executed. Subsequent commands are flushed with a syntax check only, unless FLUSH=NO.

User Response: Verify that the subject name was correctly specified. Use COPY to change the status.

DEB0173 E SUBJECT CATEGORY (subject category 'subject code') NOT DEFINED IN HIERARCHY TABLE. RELATIONSHIP WITH DIFFERENT STATUS VALUES NOT ALLOWED

Explanation: The command attempted to establish a relationship between subjects with different status codes, and one of the subjects was not defined in the DBDTHIER table. The relative positions of the subjects in the Dictionary could not be determined; the impact on the Dictionary's structural integrity and therefore could not be assessed.

System Action: The command was not executed. Subsequent commands are flushed with a syntax check only, unless FLUSH=NO.

User Response: Make an entry in the DBDTHIER table for the new subject.

DBD0174 E MORE THAN 15 SEGMENTS WERE SPECIFIED IN A STRUCTURES OUT COMMAND

Explanation: A maximum of 15 segments can be specified in a STRUCTURES_OUT command.

System Action: The command is flagged as a syntax error and is not executed.

User Response: Correct the command, and resubmit the command.

DBD0180 E INSTALLED SUBJECTS INVALID IN THIS COMMAND

Explanation: The user attempted to add or modify a relationship associated with an installed subject. The relationships and relationship data associated with an installed subject may not be modified in any way, unless that installed subject is deleted, in which case, all relationships and relationship data associated with that subject will also be deleted.

System Action: Execution of the command is terminated.

User Response: Consult the DB/DC Data Dictionary Administration and Customization Guide for rules concerning modifications to installed subjects and their associated data.

DBD0181 E CHANGES OR ADDITIONS TO INSTALLED DATA IS PROHIBITED

Explanation: The user tried to add or change the subject data associated with an installed subject. Changes or additions to subject data of an installed subject will be restricted to text data, 'DATE', and 'EXPTXT' (for categories and relationship types only).

System Action: Execution of the command is terminated.

User Response: Consult the DB/DC Data Dictionary Administration and Customization Guide for rules concerning installed subjects and their associated data.

DBD0182 E SUBJECT name CANNOT BE ADDED WITH INSTALL STATUS

Explanation: The user attempted to add a subject to CATEGORY, ATTRTYPE, or RELTYPE with (install) status (*). Control information can only be added with the 'INSTALL' command.

System Action: Execution of the command is terminated.

User Response: Consult the DB/DC Data Dictionary Administration and Customization Guide for rules concerning installed subjects.

DBD0183 E ATTRTYPE RELATED TO INSTALLED catname name

Explanation: An attribute type cannot be deleted if it is related to an installed subject category or relationship type.

System Action: Execution of the command is terminated. Attribute is not deleted.

User Response: Consult the DB/DC Data Dictionary Administration and Customization Guide for rules concerning installed subjects.

DBD0184 E CATEGORY catname INVOLVED IN INSTALLED RELTYPE relname

Explanation: The user attempted to delete an installed category which is associated with an installed relationship type. A category cannot be deleted if it is specified as the right-hand or left-hand category for an installed relationship.

System Action: Execution of the command is terminated. The category is not deleted.

User Response: Any relationship types associated with the specified category must first be deleted before the category itself can be successfully deleted. Consult the DB/DC Data Dictionary Administration and Customization Guide for rules concerning installed subjects.

DBD0185 E A SUBJECT IN CATEGORY category EXISTS IN THE DICTIONARY

Explanation: The user tried to delete a category when there were still subjects in that category in the Dictionary. This is prohibited by the Dictionary.

System Action: Execution of the command is terminated. The category is not deleted.

User Response: Delete all subjects in a category before deleting the category definition. Consult the DB/DC Data Dictionary Administration and Customization Guide for rules concerning installed subjects.

DBD0186 E INSTANCE OF RELTYPE relname EXISTS IN THE DICTIONARY

Explanation: The user tried to delete a relationship type when there were still instances of that relationship type in the Dictionary. This is prohibited by the Dictionary.

System Action: Execution of the command is terminated.

User Response: Delete all instances of the relationship type before deleting the definition. Consult the DB/DC Data Dictionary Administration and Customization Guide for rules concerning installed subjects.

DBD0187 E CANNOT CHANGE NAME OF INSTALLED catname

Explanation: The user attempted to change the name of an installed Control Information subject. The subject name or the aliases of an installed subject may not be modified.

System Action: Execution of the command is terminated.

User Response: Consult the DB/DC Data Dictionary Administration and Customization Guide for rules concerning installed subjects.

DBD0188 E UNRESTR OPTION MAY NOT BE SPECIFIED FOR DELETE OF INSTALLED SUBJECT

Explanation: The user specified the 'UNRESTR' option on the DELETE command for an installed subject. This option may cause the processor to leave installed subjects in the Dictionary which do not contain valid installation information. Therefore, the option is prohibited.

System Action: Execution is terminated.

User Response: Use the DELETE command without the option. If the installed subject has aliases, first delete the aliases, then delete the primary name. If the attribute types associated with the installed subject should also be deleted, use the DELETE_STRUCTURE command.

DBD0189 E DICTIONARY-DEFINED SUBJECTS MAY NOT BE DELETED

Explanation: The user attempted to delete one of the pre-established subjects in the Dictionary (for example, the RELTYPE category). These subjects may not be deleted.

System Action: Execution of the command is terminated.

User Response: None.

DBD IN (DBD02NN T)

DBD0201 E name DBD NOT FOUND IN LIBRARY

Explanation: The library specified by the user did not contain a DBD with this member name.

System Action: Execution is terminated for the current command, but continues with any subsequent commands.

User Response: Verify that the command was correctly specified. Verify that the DBDLIB DD statement refers to the desired library. Ensure that the DBDGEN completed successfully for this DBD.

DBD0202 E name DL/I DBS OCCURRENCE NUMBER NOT ZERO

Explanation: An attempt was made to enter a DL/I data base with a nonzero occurrence number.

System Action: Execution is terminated for the current command, but continues with any subsequent commands.

User Response: Resubmit the command with a zero occurrence number in the subject name. If an OS file is being entered, be certain that the second subject name parameter is F.

DBD0203 E name DATABASE ALREADY IN DICTIONARY CONFLICT WITH UPDATE MODE 1

Explanation: An attempt was made to enter a DBD that was already defined in the Dictionary, and update mode 1 was specified.

System Action: Execution is terminated for this command, but continues with any subsequent commands.

User Response: Verify that the command is specified correctly. If you want to add the DBD in update mode 1, use the Update commands to delete the existing data base entry.

DBD0204 W subject name SEGMENT HAS TOO MANY ALIAS FIELDS f...f DTECNTNS STRUCTURE INCOMPLETE

Explanation: The segment with this subject name has more than ten instances of fields which are contained within f...f, and an attempt was made by the program to establish a "contains within" relationship using one of the unaccommodated fields, f...f (update mode 2 or 3 only).

System Action: Execution continues with the next field in the DBD.

User Response: Use the Update commands to establish the relationships involving field f...f, if so desired. Evaluate the configuration of primary names and aliases in the Dictionary.

Execution is usually more efficient for DL/I subjects if the DL/I name is the primary name in the Dictionary. If this message appears frequently, you may wish to enlarge the table in the DBDB0000 module to accommodate more than ten entries.

DBD0205 I category name DUPLICATE NAME ADDED

Explanation: This name was added with a non-zero occurrence number. An existing entry was found with an identical subject name (update mode 1 only).

System Action: Execution continues.

User Response: Verify that such a duplicate name is desired in the Dictionary.

DBD0206 W name INDEX DATABASE INCOMPLETE

Explanation: Because of lack of Dictionary information about this index data base, the indexed data base being added may be incomplete. If the named index data base is the primary index of a HIDAM data base, the indexed data base is complete. If the named index data base is a secondary index, the indexed data base is incomplete in the Dictionary.

System Action: Execution continues.

User Response: If the named index data base is a HIDAM primary index, no action is required. If it is a secondary index, verify that the index DBD has been previously added to the Dictionary. If so, check the DBDGENs of both data bases for conflicts. After correcting any errors and/or reloading the index DBD to the Dictionary, reenter the indexed DBD using DBD_IN, specifying update mode 2 or 3, to complete the index entries.

DBD0207 E subject name SHRD INDEX ERROR - HAS ATTRIBUTES

Explanation: In handling a DL/I/VS shared index DBD under update mode 2 or 3, one of the DBD aliases was found to exist in the Dictionary with attributes, and so could not be used. No data was entered into the Dictionary.

System Action: Execution continues.

User Response: Duplicate names for DL/I DBDs are not allowed. Delete the existing name in the Dictionary or change the name in the incoming

multiple index DBD.

DBD0208 E subject name INDEX ALIAS ALREADY IN DICTIONARY. CONFLICT WITH UPDATE MODE 1

Explanation: An attempt was made to add a shared index DBD under update mode 1, and one of the index names was found to exist in the Dictionary.

System Action: Execution continues with the next input command.

User Response: Verify that the command is specified correctly. If you want to add the DBD under update mode 1, use the command language to delete the existing data base entry.

DBD OUT (DBD03NN T)

DBD0301 W name DATA BASE NOT FOUND IN DICTIONARY

Explanation: A command requested a DBDGEN output for this data base, but it could not be found in the Dictionary.

System Action: Execution ends for this command, but any subsequent commands are executed.

User Response: Verify that the command was correctly specified, especially the status (production/test) and DBD type (physical/logical) codes. If necessary, use DBD_IN to add the data base to the Dictionary.

DBD0302 W name SECONDARY INDEX INCOMPLETE IN DICTIONARY

Explanation: Certain essential attributes are missing from the Dictionary entry for this index data base.

System Action: Execution continues.

User Response: Update the Dictionary to complete secondary index specification. This can be done by repairing the DBDGEN source produced by this run, ensuring that the index DBD is in the Dictionary, then running DBD_IN against the new DBD specifying update mode 2 or 3.

DBD0303 W name SEGMENT HAS NO ATTRIBUTES

Explanation: No attribute data was found in the Dictionary for this segment.

System Action: A SEGM statement is produced with invalid or insufficient attribute keywords. Execution continues.

User Response: Repair the DBDGEN source produced, then run DBD_IN against the new DBD, specifying update mode 2 or 3, to fix the Dictionary entry.

DBD0304 W name DATA ELEMENT HAS NO 'DL/I' ALIAS

Explanation: This data element is a Dictionary primary name without an alias that is a DL/I name.

System Action: Execution continues.

User Response: Repair the DBDGEN source produced. Use the Update commands to provide the data element with the proper alias.

DBD0305 W name DATA ELEMENT HAS NO ATTRIBUTES

Explanation: No attribute data was found in the Dictionary for this data element. A SEGM statement is produced with an invalid or insufficient attribute keyword.

System Action: Execution continues.

User Response: Repair the DBDGEN source produced, then run DBD_IN against the new DBD, with mode 2 or 3.

DBD0306 W DATASET GROUP name FOR THIS DBS IS NOT IN THE DICTIONARY

Explanation: A segment was described in the Dictionary as belonging to the named data set group, but no information about that data set was found in the Dictionary.

System Action: A dummy data set statement is produced and execution continues.

User Response: Correct the DSG number for the segment following the dummy statement, if invalid; if valid, enter proper information about the DSG; for example, by repairing the DBD, executing DBDGEN, and running DBD_IN with mode 2 or 3.

DBD0307 I END OF DBD_OUT PROCESSING FOR subject name

Explanation: Processing of the current DBD_OUT command has completed.

System Action: Execution continues.

User Response: None.

DBD0308 I fieldname DATA TYPE 'type' INCORRECT FOR DBDGEN - CHANGED TO type

Explanation: The data type specified for the field would cause a failure if processed by DBDGEN. It has been changed to the valid type noted in the

message.

System Action: Execution continues.

User Response: None.

DBD0310 I FIELD xxxxx IGNORED - DEFINED AS BIT-STRING DATA - NOT SUPPORTED BY DBDGEN

Explanation: Bit-string data is not supported by DBDGEN. DBD_OUT execution ignores the field.

System Action: Execution continues.

User Response: None.

PSB IN (DBD04NN T)

DBD0401 E name PSB NOT FOUND IN LIBRARY

Explanation: The library specified by the user did not contain a PSB with this member name.

System Action: Execution is terminated for the current command; any subsequent command is executed.

User Response: Verify that the command was correctly specified, that the PSBLIB DD statement refers to the desired library, and that the PSBGEN completed successfully.

DBD0402 W name SEGMENT HAS NO 'DL/I' ALIAS

Explanation: This segment name has no alias that is a DL/I name. The segment is connected to one of the data bases referred to in the PSB.

System Action: Execution continues. The named segment is not checked against a SENSEG of the PCB being processed.

User Response: If the PSB is successfully added to the Dictionary, no action is necessary. However, the lack of a DL/I name should be investigated. If message DBD0404 appears, this lack of a DL/I name may be an explanation for the missing segment.

DBD0403 E name DATABASE NOT FOUND IN DICTIONARY

Explanation: This data base was referred to by a PCB of the PSB being added, but an entry for it was not found in the Dictionary.

System Action: The PSB is compared with Dictionary entries and any errors are noted, but the Dictionary is not changed. Any subsequent commands are processed.

User Response: Verify that the command was properly specified. Verify also that the PSB has the correct DBD names specified. If necessary, add the missing data base using DBD_IN or the Update commands, and reenter the PSB.

DBD0404 E name SEGNAME NOT FOUND IN
DATA BASE OR NOT IN
HIERARCHICAL ORDER

Explanation: This segment is specified as a SENSEG in this PSB but does not appear in the data base specified in the PSB; or the hierarchy of the PSB does not map to the hierarchy of the data base.

System Action: The remainder of the PSB is compared with Dictionary entries. Errors are noted, but the Dictionary is not changed. Any subsequent commands are processed.

User Response: Verify that the command was properly specified. Ensure that the PSB and DBDs it refer to are consistent, and that the PSBGEN and DBDGENs have the current DBD information. Make any required changes and reenter the PSB using PSB_IN.

DBD0405 I name PSB NOW IN DATA
DICTIONARY

Explanation: The PSB in the command is completed, and all related entries have been added to the Dictionary data bases.

System Action: Execution continues.

User Response: None.

DBD0406 W field VALUE EXCEEDS
DICTIONARY LIMITS - SET TO
0

Explanation: The field named had a value valid for DL/I but larger than the Dictionary limit (32767) for IOASIZE or SSASIZE.

System Action: The field value is set to 0, and execution proceeds.

User Response: Change the field to an allowable value. For a larger value, you may have to change the PSB_OUT output manually.

PSB OUT (DBD05NN T)

DBD0501 W name PSB NOT FOUND IN
DICTIONARY

Explanation: A command requested a PSBGEN output for this PSB, but it could not be found in the Dictionary.

System Action: Execution ends for this command, but any subsequent commands are executed.

User Response: Verify that the command was correctly specified, especially the status (production/test) and PSB type code. If necessary, use PSB_IN to add the PSB to the Dictionary.

DBD0502 W name PCB HAS NO RELATED
DATA BASE

Explanation: This PCB was not connected to any Dictionary entry for a data base.

System Action: A PCB statement is produced with an invalid data base name, and execution continues.

User Response: Use the Dictionary Update commands to connect the PCB to the proper data base entry. Run PSB_OUT again, or manually repair the PSBGEN source.

DBD0503 I END OF PSB_OUT PROCESSING
FOR name

Explanation: The requested PSB has been sent to the requested destination.

System Action: Processing continues.

User Response: No response is required.

DBD0504 W key PCB HAS NO ATTRIBUTES
- TYPE=DB IS ASSUMED

Explanation: The PCB with this key had no attributes in the Dictionary. For further execution, the program assumed TYPE=DB. Other error messages may follow this one.

System Action: Execution continues.

User Response: Use the ADD command to add the necessary attributes, and rerun PSB_OUT.

DBD0505 W MORE THAN number PCBs WERE
FOUND, name WILL NOT BE
PUNCHED OUT

Explanation: The table defined in module DBDQ0000 to sort PCBs is not large enough to contain all the PCBs found for the PSB_OUT that was requested. This PCB will not appear in the PSB produced.

System Action: Processing continues.

User Response: If you wish the PSB to contain all the PCBs, the table length defined by label PCBsvMAX in module DBDQ0000 must be increased.

REPORT (DBD06NN T)

**DBD0600 E PAGE OVERFLOW - RPTPAGE
MACRO MISSING**

Explanation: A required MACRO is missing in a Dictionary Report module.

System Action: Execution is terminated.

User Response: This should not occur. Notify your IBM representative.

**DBD0602 W ERROR IN TIME SVC. NOTIFY
OPERATOR**

Explanation: A system timer error has occurred.

System Action: Execution is terminated.

User Response: Notify your IBM representative.

**DBD0605 W INVALID (RPT)NAME
SPECIFIED, REPORT ON
SUBJECT FOLLOWS**

Explanation: A REPORT command with an invalid NAME= or RPTNAME= specification has been detected.

System Action: A standard report on the specified subject is produced.

User Response: If a special report is required, correct the NAME= or RPTNAME= parameter, and resubmit the command.

**DBD0606 E category SPECIFIED, BUT
SUBJECT IS NOT A category**

Explanation: A request for a report on a system object of the specified category was received and routed to DBDRDESC, but the subject is not the right category.

System Action: Execution of the report command is terminated.

User Response: Correct the command, and resubmit it.

DBD0608 I DESC VALUE DEFAULTED TO 1

Explanation: A DESC= value was not supplied with the command and the default of 1 has been used.

System Action: Execution continues.

User Response: None.

**DBD0610 S A DL/I ERROR HAS BEEN
FOUND WHILE ATTEMPTING TO
PROCESS A REQUEST FOR AN
ISR REPORT**

Explanation: A DL/I error has been found while attempting to process a request for an ISR report.

System Action: Processing of the Report command is terminated.

User Response: Consult the related DL/I messages and codes manual for the returned status code. If a data base error is indicated, correct the data base.

**DBD0611 E SUBJECT TYPE IS NOT 'E',
'S', OR 'D'**

Explanation: This message was issued when an invalid subject was specified for an Indirect Subject Reference report. This report only scans keywords which contain references to element, segment, or data base subjects. Therefore, subjects in categories other than these will not have references which can be reported using this command.

System Action: Processing of the Report command is terminated.

User Response: If references to the subject may be in User Data or Description, use the SCAN command to report it.

**DBD0612 E "DETAIL" VALUE parameter
INVALID FOR SUBJECT TYPE
subject type**

Explanation: A DETAIL value was found that is not valid for the specified subject type.

System Action: REPORT command execution is terminated.

User Response: Specify valid values, and resubmit the command.

**DBD0614 E NAME FOR ISR REPORT
GREATER THAN 8 CHARACTERS**

Explanation: The name specified in the REPORT command has more than eight characters.

System Action: Execution of the REPORT command is terminated.

User Response: Resubmit the command with the correct name of eight or fewer characters.

**DBD0615 E REPORT SUBJECT NOT
SPECIFIED IN COMMAND**

Explanation: A report command was entered without specifying the subject category and the subject name. These are required for all reports, except the glossary reports.

System Action: Execution of the command is terminated.

User Response: Correct the command, and resubmit it.

DBD0620 W DL/I ERROR ON DATABASE
name STATUS WAS status
LAST SEG WAS segment
SUBJECT WAS subject

Explanation: A DL/I error return code was encountered while retrieving information from a Dictionary data base.

System Action: Execution for the subject that caused the error is terminated. Execution continues for any subjects remaining in the Dictionary data base.

User Response: Correct the condition responsible for the bad return code and resubmit.

DBD0621 E NO SUBJECT CAN SATISFY
BOTH THE TO AND FROM
OPERANDS

Explanation: The user-name specified in the "TO" operand occurs before the user-name specified in the "FROM" operand. No subjects can satisfy the criteria specified in the input command.

System Action: Execution of the command is terminated.

User Response: Correct the "TO" and/or "FROM" operand in the command, and resubmit the command for execution.

DBD0622 E ERROR ENCOUNTERED DURING
SORT OF REPORT NAMES

Explanation: The sort routine encountered an error that prevents it from completing successfully. The error can include an out-of-sequence condition or an incorrectable I/O error. (See OS/VS Sort/Merge Programmer's Guide or DOS/VS Sort/Merge Programmer's Guide for more information.

System Action: Execution of the command is terminated. No report is generated.

User Response: Consult the SORT routine documentation for possible problems. Correct the error, and resubmit the job.

DBD0623 E SORT KEYWORD IS INVALID IN
ONLINE MODE

Explanation: The report sort option cannot be used while running on line.

System Action: Command execution is terminated. No report is generated.

User Response: If a sorted report is desired, submit the command in batch mode. A report command without the

sort option is valid on line.

DBD0630 I CATEGORY category-name
DOES NOT EXIST OR IS NOT
INSTALLED

Explanation: category-name has been specified where only an installed category name is acceptable. This message may be issued when requesting a Guide or Format report or when specifying RELTYPES as a DETAIL= operand on a REPORT command. However, category-name is not defined in the CATEGORY category or exists but is not in installed status.

System Action: If a Format or Guide report was requested, there is no further processing on this command. Otherwise, the operand is ignored and processing continues.

User Response: Consult your data base administrator for valid category names.

DBD0631 E RELATIONSHIP TYPE reltype
DOES NOT EXIST OR IS NOT
INSTALLED

Explanation: reltype has been specified where only an installed relationship type is acceptable. However, reltype is not defined in the RELTYPE category or exists but is not in installed status.

System Action: There is no further processing on this command.

User Response: Consult your data base administrator for valid relationship-type names.

DBD0632 E UNEXPECTED RETURN CODE
code FROM PROGRAM ACCESS
FACILITY IN MODULE
module-name

Explanation: An error condition was detected by the program access facility while retrieving data for the report from the dictionary data bases. The specific error that was detected can be found by checking the particular code returned in program access facility return codes. This message usually indicates that the dictionary data bases are in error or a program error occurred.

System Action: There is no further processing on this command.

User Response: Contact your IBM representative for assistance.

DBD0633 E AN INVALID CATEGORY NAME
WAS SPECIFIED FOR A
report-type REPORT

Explanation: Only the categories RELTYPE or CATEGORY are allowed on a Guide or Format type report.

System Action: There is no further processing for this command.

User Response: Specify either RELTYPE or CATEGORY for the category name.

DBD0634 I THERE IS NO RELTYPE WITH
CATEGORY = category-name,
KEYWORD = rel-keyword, AND
RELATED CATEGORY =
category-name2

Explanation: There is no relationship type that is defined for category-name, rel-keyword, and category-name2.

System Action: The operand specifying this relationship type is ignored.

User Response: Check the categories and relationship keyword, and specify correct values.

DBD0635 E GETMAIN/GETVIS UNABLE TO
GET number BYTES OF
VIRTUAL STORAGE IN MODULE
module-name

Explanation: A GETMAIN (OS) or GETVIS (DOS) was attempted but failed because not enough virtual storage was available.

System Action: There is no further processing on this command.

User Response: Take the action necessary to make more virtual storage available, such as increasing the region or partition size, and reexecute the command.

DBD0680 E subject name IS NOT AN
INDEX DATA BASE

Explanation: The data base that was referenced is not an index data base.

System Action: The command is flushed.

User Response: Correct the data base name, and resubmit the request.

DBD0681 I DBS data base name HAS NO
RELATED SEGMENTS

Explanation: A report of related segments was requested and none exist. This message is issued instead of a blank report form.

System Action: Execution continues.

User Response: No response is required.

DBD0689 I subject category subject
name DOES NOT EXIST

Explanation: The subject name could not be found.

System Action: No report is produced.

User Response: Verify that the subject category and subject name were correctly specified.

STRUCTURES OUT (DBD07NN T)

DBD0701 E COULD NOT OBTAIN PRIMARY
SEGMENT FOR segname

Explanation: STRUCTURES_OUT was requested to build a structure for a given segment. The segment whose name was given could not be found in the Dictionary data base.

System Action: STRUCTURES_OUT terminates execution for the requested segment.

User Response: Correct the segment name, and resubmit the request.

DBD0702 E COULD NOT OBTAIN DYNAMIC
STORAGE REQUIRED FOR
INTERNAL BLOCK. STORAGE
MANAGER RETURN CODE=nnnn.

Explanation: A call was made to the Storage Manager, DBDSSTOR, and resulted in a nonzero return code indicating that no storage was obtained. Either free storage was not available, or an error occurred within GETMAIN (OS systems) or GETVIS (DOS systems).

System Action: The function being processed is terminated. Return code=8.

User Response: If the return code indicates a system error (nnn is the return code from GETVIS (DOS) or GETMAIN (OS)), notify your IBM representative. If the return code indicates that insufficient storage was available, increase the amount of storage assigned to the region, and rerun the command.

DBD0703 W LENGTH GIVEN FOR SEGMENT
segname IS LESS THAN THE
TOTAL LENGTH OF ITS
SUBSTRUCTURE. STRUCTURE
OUT MAY BE INCORRECT.

Explanation: The length of the segment as given in its SEGATRB segment is less than the sum of the lengths of the elements subordinate to it in the structure.

System Action: STRUCTURES_OUT makes no adjustments, nor does it attempt to create any fillers. If the segment in question was part of a MULTISEG request, the output structure could be incorrect.

User Response: Inspect the structure definition in the Dictionary and change it as necessary to make the

lengths consistent.

**DBD0704 E ERROR READING SEG DTE FOR
SEGMENT segname, DL/I
STATUS CODE: cc**

Explanation: The DL/I status code indicates the type of error.

System Action: STRUCTURES_OUT terminates execution for this segment.

User Response: Correct the error in the data base.

**DBD0705 E SEGMENT segname HAS NO
ELEMENTS**

Explanation: STRUCTURES_OUT could not find any SEG DTE segments under the requested segment.

System Action: STRUCTURES_OUT terminates execution for the current request. Return code=8.

User Response: Complete the structure, using ADD_RELATIONSHIP commands.

**DBD0706 W ERROR READING SGPLINFO FOR
SEGMENT segname, DL/I
STATUS CODE: cc**

Explanation: DL/I returned a nonblank status code other than GE.

System Action: STRUCTURES_OUT continues execution. Comments may be missing from the segment level data definition, or if PL/I was the requested language, data attributes may be omitted.

User Response: Correct the error in the data base.

**DBD0707 W ERROR READING SEGSDRY FOR
SEGMENT segname, DL/I
STATUS CODE cc**

Explanation: DBDSSEG was attempting to find an alias appropriate to the requested language. An error status code was sent by DL/I.

System Action: STRUCTURES_OUT ignores the error and uses the primary name as the segment name in the output structure. Return code=4.

User Response: Determine the type and cause of the error in the data base, correct it, and resubmit the SO command.

**DBD0708 E DTE dtename NAMED IN
SEG DTE UNDER SEG segname
NOT FOUND**

Explanation: DBDSPATH was attempting to read the named DTE and received a status code indicating not found from DL/I.

System Action: STRUCTURES_OUT terminates execution for the current request.

User Response: Add the named DTE to the data base.

**DBD0709 E ERROR READING DTE SEGMENT
FOR dtename, DL/I STATUS
CODE: cc**

Explanation: DBDSPATH was attempting to read the named DTE and an error status code was returned by DL/I.

System Action: STRUCTURES_OUT terminates execution for the current request.

User Response: Determine the type and cause of the error, correct the data base, and resubmit the SO command.

**DBD0710 E ERROR READING DTECNTNW
SEGMENT UNDER DTE**

Explanation: DBDSPATH was attempting to read the DTECNTNW segment and received an error status code from DL/I.

System Action: STRUCTURES_OUT terminates execution for the current request.

User Response: Determine the type and cause of the error, correct the data base, and resubmit the SO command.

**DBD0711 E COULD NOT OBTAIN DTE
SEGMENT dtename CONTAINED
WITHIN dtename, DL/I
STATUS CODE: cc**

Explanation: DBDSPATH was attempting to read a DTE segment named in a DTECNTNW segment. DL/I returned a status code indicating either an error or not found.

System Action: STRUCTURES_OUT terminates execution for the current request.

User Response: If an error was detected, correct the data base, and resubmit the SO command. If the DTE was not found, add it to the data base, and resubmit the SO command.

**DBD0712 E ERROR READING DTESEG
SEGMENT FOR DTE dtename,
DL/I STATUS CODE: cc**

Explanation: DBDSPATH was attempting to read a DTESEG segment for the named DTE. DL/I returned an error status code.

System Action: STRUCTURES_OUT terminates execution for the current request.

User Response: Determine the type and cause of the error, correct the data base, and resubmit the SO command.

DBD0713 W REDEFINING DTE FOUND AND REQUESTED LANGUAGE IS PL/I. ELEMENT dtename AND ITS SUBSTRUCTURE WERE NOT PROCESSED

Explanation: The PL/I DEFINED attribute is not supported in the current level of the program product.

System Action: Processing continues.

User Response: No response is required. STRUCTURES_OUT merely ignores the named element and any elements subordinate to it. The basic data structure is produced correctly; only the redefining elements are omitted.

DBD0714 W COULD NOT OBTAIN DTEATRB SEGMENT FOR DTE dtename. DL/I STATUS CODE: cc

Explanation: A status code indicating either an error or not found was returned by DL/I when STRUCTURES_OUT attempted to read the named DTEATRB segment.

System Action: STRUCTURES_OUT assumes a length of zero.

User Response: If it is an error, determine the type and cause of the error, correct the data base, and resubmit the SO command. If the code indicates not found, add the length and type attributes to the element in the data base, and resubmit the SO command.

DBD0715 W LENGTH GIVEN FOR DTE dtename IS LESS THAN THE TOTAL LENGTH OF ITS SUBSTRUCTURE. STRUCTURE OUT MAY BE INCORRECT

Explanation: The length of the element as given in its DTEATRB segment is less than the sum of the lengths of the elements subordinate to it in the structure.

System Action: STRUCTURES_OUT will make internal adjustments in the length for the purpose of producing a proper structure. This structure may not be equivalent to the actual segment in the user data base.

User Response: Verify that the output structure is indeed equal to the segment in the data base. If it is not, check the lengths of the elements involved, and correct the length attributes to match the segment in the user data base.

DBD0716 W DTE dtename-1 REDEFINES dtename-2 WHICH COULD NOT BE FOUND AT THE SAME LEVEL IN THE STRUCTURE

Explanation: Dtename-1 was found to contain dtename-2 with a relationship of REDEFINES (DTEUID=R). Dtename-2 was not contained within the same element or segment as dtename-1.

System Action: STRUCTURES_OUT ignores this relationship and continues processing.

User Response: If this is an error, respecify the structure so that both elements are at the same level in the structure, and contained within the same superior entity (segment preceding lower-level group item).

DBD0717 E COULD NOT OBTAIN DTE SEGMENT FOR dtename. DL/I STATUS CODE cc.

Explanation: DL/I returned a status code indicating either an error or not found when STRUCTURES_OUT attempted to read the named DTE.

System Action: STRUCTURES_OUT terminates execution for the current request.

User Response: Determine the type and cause of the error, correct the Dictionary data base, and resubmit the SO command.

DBD0718 E COULD NOT OBTAIN DTEVALUE SEGMENT FOR LEVEL-88 DTE dtename. DL/I STATUS CODE: cc

Explanation: DL/I returned a status code indicating either an error or not found when STRUCTURES_OUT attempted to read the DTEVALUE segment for the named DTE.

System Action: STRUCTURES_OUT terminates execution for the current request.

User Response: Determine the type and cause of the error, correct the Dictionary data base, and resubmit the SO command.

DBD0719 W STARTING POSITION FOR A REDEFINING DTE (dtename) IS NOT EQUAL TO THE START POSITION FOR THE PREVIOUS ELEMENT. THE STRUCTURE OUT MAY BE INCORRECT

Explanation: In processing a redefining element, STRUCTURES_OUT found that the starting position did not match that of the immediately preceding element at the same level.

System Action: STRUCTURES_OUT makes internal adjustments of the starting position to allow it to produce a proper structure. No changes are made in the Dictionary. This adjustment procedure may cause further warnings to be generated.

User Response: Correct the start position of the redefining element to match that of the element it redefines, and resubmit the S0 command.

DBD0720 W THE LENGTH OF THE REDEFINING DTE (dtename) IS GREATER THAN THAT OF THE ELEMENT IT REDEFINES. STRUCTURE OUT MAY BE INCORRECT

Explanation: The length of the named DTE is greater than the length of the immediately preceding element at the same level.

System Action: STRUCTURES_OUT makes internal adjustments in the length of the preceding element which do not appear in the output structure (or in the Dictionary). The output structure may not be equivalent to the actual segment in the user data base.

User Response: Either correct the length attribute of the redefining element or reverse the redefining relationship so that the shorter element redefines the longer.

DBD0721 W THE START POSITION OF A NON-REDEFINING ELEMENT (dtename) IS LESS THAN THE MINIMUM START POSITION CALCULATED BY STRUCTURES OUT. STRUCTURE OUT MAY BE INCORRECT.

Explanation: STRUCTURES_OUT has found that with the given start position the named element would overlay part or all of the previous element (or the beginning of the segment). Since a redefining relationship was not specified, a potential error exists.

System Action: STRUCTURES_OUT makes internal adjustments to eliminate the overlapping condition (no change is made in the Dictionary data base). The resulting structure may not be equivalent to the segment in the user data base.

User Response: Either correct the starting position, enter a REDEFINES relationship, or respecify the structure to eliminate the overlap.

DBD0722 W COULD NOT OBTAIN SEGATRB SEGMENT FOR segname, DL/I STATUS CODE: cc

Explanation: DL/I returned a status code indicating either an error or not found when STRUCTURES_OUT attempted to read the named SEGATRB segment. Both conditions constitute an error from the viewpoint of STRUCTURES_OUT.

System Action: STRUCTURES_OUT assumes a length of zero.

User Response: Determine the type and cause of the error, correct the Dictionary data base, and resubmit the S0 command.

DBD0723 E ERROR FINDING PRIMARY NAME FOR subject-name. DBDGPRIM RETURN CODE: nn

Explanation: STRUCTURES_OUT called DBDGPRIM to obtain the primary name for the named subject. DBDGPRIM returned a nonzero return code. A return code of 1 means that the PCB status code is other than blank or GE, indicating an unrecoverable DL/I error has occurred. A return code of 3 indicates that the PCB status code is GE and the root segment was not found. Therefore, the given key is invalid.

System Action: STRUCTURES_OUT terminates execution for the current request.

User Response: From the S0 command, determine the data base involved, correct the data base, and resubmit the S0 command.

DBD0724 E COULD NOT OBTAIN PRIMARY SEGMENT FOR subject-name. DL/I STATUS CODE: cc

Explanation: DL/I returned a status code indicating either an error or not found when STRUCTURES_OUT attempted to read the primary segment for the named subject.

System Action: STRUCTURES_OUT terminates execution for the current request.

User Response: From the S0 command, determine the Dictionary data base involved, correct the data base, and resubmit the S0 command.

DBD0725 E ERROR READING xxxSEG RELATIONSHIP SEGMENT FOR subject-name. DL/I STATUS CODE: cc

Explanation: DL/I returned a status code indicating either an error or not found when STRUCTURES_OUT attempted to read the named relationship record. xxx is the data base name taken from the subject code in the S0 command.

System Action: STRUCTURES_OUT terminates execution for the current request.

User Response: Determine the type and cause of the error, correct the Dictionary data base involved, and resubmit the S0 command.

DBD0726 I STRUCTURES OUT COMPLETE

Explanation: STRUCTURES_OUT has finished processing one request with no severe errors.

System Action: STRUCTURES_OUT returns to its caller.

User Response: No user response is required.

DBD0727 E subject-name HAS NO RELATED SEGMENTS

Explanation: The named subject was found to have no segments related to it.

System Action: STRUCTURES_OUT terminates execution for the current request.

User Response: Complete the structure by adding the appropriate subject-to-segment relationships.

DBD0728 I MEMBER NAME DEFAULTED TO 'NONAME'

Explanation: MEMBER= keyword was not specified; 'NONAME' was assumed.

System Action: For an OS system the ./ ADD card will read: ./ ADD NAME=NONAME. In DOS, the CATALS card will appear as follows: CATALS [A|P|C] NONAME.

User Response: Specify MEMBER= keyword.

DBD0729 I LANGUAGE DEFAULTED TO ASSEMBLER

Explanation: The LANG= keyword was omitted from the S0 command.

System Action: The structure will be produced in assembler language statements.

User Response: No user response is required.

DBD0730 E FOR ELEMENT (element-name), THE LENGTH (length) TIMES THE OCCURS COUNT (occurs count) EXCEEDS THE MAXIMUM ALLOWABLE SEGMENT LENGTH

Explanation: The total space to be allocated for all occurrences of the element exceeds 32767, the maximum length possible in the Dictionary.

User Response: Either the length or the occurs count is too large. Correct the erroneous data, using the

CHANGE_IN command, and rerun the S0 command.

DBD0731 W SEGMENT/ELEMENT subject name HAS ZERO LENGTH. A LENGTH OF 1 IS ASSUMED.

Explanation: Either no length attribute was supplied when the segment or element was created or BYTES=0 was specified. STRUCTURES_OUT assumes a length of 1 and continues execution. This may cause message DBD0703 to appear.

User Response: Add the proper length attribute to the segment or element, and rerun STRUCTURES_OUT.

DBD0732 E THE LENGTH OF ELEMENT element-name CAUSES THE CURRENT STRUCTURE TO EXCEED THE MAXIMUM ALLOWABLE SEGMENT LENGTH.

Explanation: STRUCTURES_OUT has determined that the cumulative length of the structure (including the length of the named element) would exceed 32767 bytes, the maximum length that can be stored in the Dictionary.

System Action: Processing of the structure is terminated.

User Response: Examine all element lengths which appear in the structure prior to (and including) the named element. Correct any erroneous lengths and/or occurs counts and rerun STRUCTURES_OUT.

DBD0733 W ELEMENT element-name, WHOSE DATATYPE=x AND LENGTH=length, IS NOT TRANSLATABLE TO USAGE COMP. USAGE DEFAULTS TO DISPLAY.

Explanation: STRUCTURES_OUT was attempting to produce a COBOL data definition for which no COBOL attributes were found in the Dictionary. A data type x(DTETYPE=x) was found and the length was not 2, 4, or 8. A conflict between usage DISPLAY and a numeric VALUE clause may result.

System Action: Processing continues using a picture of DISPLAY.

User Response: Either change the length to 2, 4, or 8, or change the data type to conform to the intended COBOL usage.

DBD0734 I COMPRESS OPTION DEFAULTED TO 'NO'

Explanation: The COMPRESS parameter was not specified in a STRUCTURES_OUT or RECALCULATE_SEGMENT command, and the default value of NO has been used.

System Action: Command execution continues.

User Response: If you wish to have your output structure compressed (fillers deleted), resubmit the command and specify COMPRESS=Y.

DBD0735 I DVALUES OPTION DEFAULTED TO 'NO'

Explanation: The DVALUES parameter was not specified in the STRUCTURES_OUT command, and the default value of NO has been used.

System Action: Command execution continues.

User Response: If you wish to have DVALUES data included in your output, resubmit the command and specify DVALUES=Y.

DBD0736 I NARRATIVE OPTION DEFAULTED TO 'YES'

Explanation: The NARATIV keyword was not specified in a STRUCTURES_OUT command, and the default value of YES has been used.

System Action: Command execution continues.

User Response: If you wish that narrative data be omitted from the output structure, resubmit the command and specify NARATIV=N.

DBD0737 I LEVEL OPTION DEFAULTED TO '1'

Explanation: The LEVEL keyword was not specified in a STRUCTURES_OUT command, and the default value of '1' has been used.

System Action: Command execution continues.

User Response: If you wish the structure segment level to be other than 1, resubmit the command and specify LEVEL=n (n may be any number from 1 to 255).

DBD0738 I SEGATR OPTION DEFAULTED TO 'YES'

Explanation: The SEGATR keyword was not specified in a STRUCTURES_OUT command for a PL/I structure, and the default value of YES has been used.

System Action: Command execution continues.

User Response: If you wish the structure to not use the installation option, resubmit the command and specify SEGATR=N or SEGATR='data string' (a maximum of 30 characters may be used in the latter format).

DBD0739 W SEGMENT name, CONTAINING A MULTIPLY RELATED FIELD, name, IS MISSING AT LEAST ONE SEGTE RELATIONSHIP - POSSIBLE ERROR IN STRUCTURES_OUT OR RECALCULATE_SEGMENT

Explanation: The following situation was encountered: a segment contains multiple fields with the same name, and some, but not all, of them are related to the segment. All first level occurrences have been related to the segment, but not all the non-first level occurrences have been.

System Action: Processing continues.

User Response: If this situation has caused errors, then Dictionary update commands should be used to correct the problem.

DBD0740 E LVL TWO VALUE LESS THAN LEVEL VALUE, PROCESSING TERMINATED

Explanation: The value specified for the LVL TWO parameter is greater than the value specified for the LEVEL parameter.

System Action: Execution is terminated for this command.

User Response: Change the parameters used on the invocation of the STRUCTURES_OUT command.

DBD0741 E NEXT LEVEL NUMBER (\$) IS INVALID, PROCESSING TERMINATED

Explanation: In execution of a COBOL structure, a level number greater than 49 would be produced if execution continued. This is a COBOL restriction.

System Action: Execution is terminated for this command.

User Response: Change the parameters used on the invocation of the STRUCTURES_OUT command (for example, specify a lower LVLINCR value).

DBD0742 W user usage INCONSISTENT WITH GENERATED PICTURE FOR FIELD field, generated usage USED

Explanation: During Structures-out execution for a RECALCULATE_SEGMENT command, a field was found which had a COBOL USAGE parameter specified, and did not have a COBOL PICTURE parameter specified. Structures-out generates PICTURE and USAGE clauses based on the TYPE and BYTES values stored when PICTURE is not stored. This message will occur when the generated USAGE clause does not match the one stored in the Dictionary.

System Action: The generated USAGE is used and execution continues.

User Response: The Dictionary contains conflicting data. TYPE and BYTES do not match the USAGE value. The user probably should add a PICTURE clause and make the TYPE, BYTES, and USAGE values consistent.

DBD0750 I SEGATR=YES EITHER SPECIFIED OR DEFAULTED BUT NO ATTRIBUTES DEFINED BY INSTALLATION

Explanation: PL/I attributes are to be produced by STRUCTURES_OUT for the segment level definition. However, the installation has not defined these default attributes.

System Action: Execution continues; the structure will be produced as if SEGATR=NO had been specified.

User Response: Define the attributes via the 'DEFAULT' facility, or specify the desired attributes as a character string following the keyword.

DBD0760 W SEGMENT/ELEMENT NAME OR NAME CREATED BY PREFIX/SUFFIX OPTION IS TOO LONG FOR SPECIFIED LANGUAGE - NAME name IS TRUNCATED

Explanation: The names created for output by application of the prefix and/or suffix option has resulted in a PL/I name, COBOL name, or assembler name exceeding 31 characters, 30 characters, or 8 characters, respectively.

System Action: Execution continues.

User Response: Specify a combination of prefix, suffix, and name which will not exceed the limit for the language being used.

DBD0761 W SEG/DTE NAME (xxxx) - LENGTH (nnn) INCONSISTENT WITH DATA TYPE (x). THE STRUCTURE OUT MAY BE INCORRECT.

Explanation: A PL/I definition of the specified type must conform to the following length limitations:

TYPE	D	E	F	H	P	Z
LENGTH	8	4	4	2	1-8	1-15

System Action: Execution continues.

User Response: The length should be changed to the value, or one of the values, allowed for that type.

DBD0762 W ELEMENT xxxx WHOSE LENGTH IS nnn, TRANSLATES TO A BIT VALUE WHICH EXCEEDS

THE PL/I COMPILER LIMIT - TYPE DEFAULTS TO CHARACTER.

Explanation: Any element whose DTETYPE is hexadecimal and length is not 2 or 4 is translated to a bit definition. However, the translated value exceeds the limit of the PL/I compilers. The element has been produced with a character definition.

System Action: Execution continues.

User Response: Verify that the DTETYPE and/or LENGTH is the desired value. If not, change to the correct one.

DBD0763 W IN THE DIMENSION ATTRIBUTE PRODUCED FOR FIELD name, THE VALUE OF AT LEAST ONE LOWER BOUND EXCEEDS THAT OF ITS CORRESPONDING UPPER BOUND

Explanation: A DIMENSION attribute that has been generated based on data found in the Dictionary data bases is invalid for PL/I.

System Action: The invalid DIMENSION is produced and processing continues.

User Response: Change the upper or lower dimension value to produce a valid set of values.

DBD0764 W NEGATIVE SCALE NOT SUPPORTED

Explanation: The precision attribute for the element currently being processed contains a negative scale. Negative scale is not supported when attempting to generate a PICTURE clause.

System Action: The scale value is assumed to be zero.

User Response: Correct the error, or use the assumed value.

DBD0765 W SCALE MAY NOT BE GREATER THAN THE LENGTH

Explanation: The precision attribute for the element currently being processed contains a scale value greater than the total length.

System Action: The scale value is assumed to be zero.

User Response: Correct the error.

DBD0790 I RECALCULATION OF SEGMENT segment TERMINATED

Explanation: Execution of the RECALCULATE_SEGMENT command was terminated because of the error condition indicated by a preceding message.

System Action: RECALCULATE_SEGMENT command execution is terminated. Control is returned to process the next user input.

User Response: Take whatever steps are necessary to correct the error indicated in the preceding message.

DBD0798 I NO CHANGES TO THE SEGMENT STRUCTURE REQUIRED

Explanation: In execution of the RECALCULATE_SEGMENT command, it has been determined that no changes are required in the segment structure.

System Action: Execution continues.

User Response: None.

DBD0799 I RECALCULATION OF SEGMENT segment COMPLETED

Explanation: The RECALCULATE_SEGMENT command has completed execution successfully.

System Action: Control is returned to process the next user input.

User Response: None.

SCAN (DBD08NN T)

**DBD0801 E STRING TOO LONG - BEGINS:
X...X**

Explanation: The string that begins x...x contains more than 31 characters. A delimiter may have been misplaced. SCAN execution is bypassed.

System Action: Execution of the command is terminated.

User Response: Reenter the command, observing the maximum string size of 31.

DBD0802 E STRING TOO SHORT:

Explanation: The scan string associated with a keyword was omitted from the command.

System Action: SCAN execution is bypassed.

User Response: Reenter the command, observing the minimum string size of 1.

DBD0803 E STRING HAS NO ENDING APOSTROPHE: xx...x

Explanation: The end of the command was reached, or 31 characters were exceeded, without encountering an ending apostrophe for string xx...x.

System Action: SCAN execution is bypassed.

User Response: Reenter the command with an ending apostrophe.

DBD0804 I NO SUBJECT CATEGORY FOUND, DTE ASSUMED.

Explanation: No subject category was found on the SCAN command.

System Action: The subject category defaults to DTE.

User Response: None.

DBD0805 I RELTYPE=SUBJECT CATEGORY, NAME SCAN IMPLIED

Explanation: The processor found a request for a scan of related subjects of the same subject category as is being scanned (for example, SCAN DTE REL=(DTE,xxx)). This was assumed to imply a scan of all names and aliases in the subject category. (For system type subject categories, PSB, TRN, PGM, JOB, MOD, and SYS, this message does not occur).

System Action: A name scan is performed, that is, SCAN DTE REL=(DTE,xxx) becomes SCAN DTE NAME=xxx.

User Response: None.

DBD0806 I LIMIT SPECIFIED EXCEEDS MAXIMUM. DEFAULTS TO nn

Explanation: A limit greater than nn was specified in the SCAN command.

System Action: The command is processed under a default limit of nn.

User Response: None.

DBD0807 I SCAN PROCESSING BYPASSED

Explanation: An error indicated in a previous message has caused bypassing of the scanning function.

System Action: SCAN execution is bypassed. Processing continues with the next command.

User Response: None.

DBD0808 I SCAN LIMIT REACHED, nn NAMES LISTED

Explanation: The user specified a scan limit, or a default value was applied and more subjects passed the scan than were allowed.

System Action: Execution ceases, but nn reports or names are produced.

User Response: To discover the names of all subjects passing the scan, the user may specify the LST option with

no limit value specified. A more stringent scan criterion might provide a more manageable set of subjects.

DBD0809 I SCAN FOUND xxx HITS

Explanation: Scan execution successfully located xxx subjects passing the scan criteria.

System Action: Execution continues.

User Response: None.

DBD0810 I SCAN PROCESSING COMPLETED

Explanation: Processing of the SCAN command has ended.

User Response: None.

DBD0811 W NO CRITERION KEYWORD WAS SPECIFIED

Explanation: No scan criterion keyword was specified.

System Action: Execution continues.

User Response: Reenter the command, specifying a criterion keyword if required.

DBD0812 E START SUBJECT SPECIFIED DOES NOT EXIST

Explanation: The START operand specifies the starting position for SCAN function. That subject must appear in the data base to be scanned.

System Action: Execution of the command is terminated.

User Response: Specify a subject that is known to be in the data base if you want to start scanning the data base from that point. Omit the START keyword if you want to begin scanning with the first subject in the data base. Resubmit the command for execution.

DBD0813 E LINENO SPECIFICATION IS INVALID

Explanation: The minimum text line number specified in the LINENO operand is greater than the maximum text line number specified. There are no text lines that can satisfy this condition.

System Action: Execution of the command is terminated.

User Response: Correct the operand, and resubmit the command.

COPY (DBD09NN T)

DBD0901 I END OF COPY

Explanation: Copy processing is complete.

System Action: Processing continues with the next command.

User Response: None.

DBD0902 I subject category (subject name) HAS BEEN COPIED

Explanation: The subject named (subject B in the COPY command or another subject in the hierarchy) has been copied successfully.

System Action: Execution continues with the next subject in the hierarchy, if any.

User Response: None.

DBD0903 W RELATIONSHIP BETWEEN subject category (subject name) AND subject category (subject name) NOT COPIED TO PRESERVE STRUCTURAL INTEGRITY

Explanation: This relationship was not copied because it would have violated the status rules.

System Action: This relationship is ignored, and execution continues.

User Response: Review the COPY results to verify that the relationship was not required.

DBD0904 E INVALID OPTION ON COPY COMMAND.

Explanation: The basic/default type of copy can be altered by the 'relate', 'dependents', or 'affected' parameters. The option chosen was invalid for one of these reasons:

- The 'affected' option is not allowed for subjects in the extensibility data base.
- The 'relate' or 'dependents' option is only allowed on Extensibility subjects residing in the 'category' or 'reltype' categories.

System Action: Execution continues with the next command.

User Response: Correct the command, and resubmit it.

DBD0906 W SUBJECT CATEGORY (subject name) contains a SEGMENT (segname, code1,code2) NOT FOUND IN HIERARCHY TABLE

Explanation: The copy function could not find 'segname' in its hierarchy table, or if it was found but is a relationship segment such as SYSxxx, xxxSYS, EXTxxx, xxxEXT, SYSSYS, or

EXTEXT, then one or both of the category codes code1 or code2) have not been entered in the table for the category involved.

This situation is most likely a Dictionary program error. However, it is possible that modification of Dictionary DBDs by an installation could cause the problem.

System Action: The segment is not copied. Execution continues with the next segment under the root of the segment currently being copied.

User Response: If you have defined additional segments in any Dictionary data base or have added subject codes to the SYSTEM data base, their descriptions and hierarchical position should be added to the hierarchy table, named HIERTBL, located in module DBDYLEVL.

The absence of a relationship segment definition in this table has the effect of treating the relationship as 'nonstructural.' When nonstructural relationships have been properly defined in the table, message DBD0911 is issued instead of DBD0906.

DBD0907 W subject category (subject name) WAS NOT COPIED AS AN ALIAS OF subject category (subject name) DUE TO CONFLICTS WITH ALIAS RULES

Explanation: The first subject name already is an alias or primary name of another entry, so it could not be made an alias.

System Action: This alias is ignored, but copy execution continues.

User Response: Add another alias with a new name, or examine the alias not copied to determine the appropriate action.

DBD0908 E SUBJECT CATEGORIES ARE NOT THE SAME

Explanation: Subject categories must be the same.

System Action: Execution continues with the next command.

User Response: Correct the command, and resubmit it.

DBD0909 E SUBJECT NAMES ARE IDENTICAL

Explanation: Subject names must not be the same. This could happen if both names coded on the COPY command were inadvertently the same. However, it more likely occurred because either subjectnameA or subjectnameB was an alias (or both were). When converted to primary names, duplicates result.

System Action: Execution continues with the next command.

User Response: Correct the command, and resubmit it.

DBD0910 I SEG (segname) WAS ADDED AS A ROOT ONLY, SO IT COULD BE PROPERLY RELATED TO PCB (pcbname)

Explanation: In copying the requested structure, a PCBSEG relationship was encountered. The relationship and any relationship data (subordinate segments) cannot be copied until a SEG subject exists. This SEG subject will normally be copied later, via the DBSSEG relationship, but a temporary 'place holder' is created now to facilitate the COPY process.

System Action: SEG subject, root only, is copied, and PCBSEG relationship is also copied.

User Response: None required.

DBD0911 W THE NON-STRUCTURAL (relkw) seqvalue RELATIONSHIP BETWEEN subject category (subject name) AND subject category (subject name) WAS NOT COPIED OR FOLLOWED.

Explanation: The first subject named in the message is related to the second subject via a relationship treated by COPY as NON-STRUCTURAL. The specified relationship is recognized, reported on via this message, but not copied. Relkw is the relationship keyword and seqvalue is the relationship sequence value, if one exists. The message can be caused by one of the following:

- A relationship exists between a subject in an installation defined category and a subject in one of the standard Dictionary categories.
- A relationship exists between two subjects in the same category within the SYSTEM data base.
- A relationship exists between subjects in standard Dictionary categories that is not supported by the Dictionary hierarchy table shown in the DB/DC Data Dictionary Applications Guide. The possibilities follow:
- A relationship exists between a MODULE subject and a TRANSACTION subject.
- A relationship exists between a SYSDEF subject and a JOB, SYSTEM, MODULE, or PROGRAM subject.

System Action: The relationship is not copied.

User Response: If the relationship applies to the newly-created copy of the subject, it must be manually reconstructed.

ADD (DBD10NN T)

DBD1001 I subject category subject name ALREADY EXISTS

Explanation: The subject name could not be added because it already exists.

User Response: Verify that the subject category and subject name were correctly specified.

DBD1002 I subject category subject name HAS BEEN ADDED

Explanation: The command was executed as requested.

System Action: Execution of the command has terminated successfully.

User Response: No response is required.

CHANGE IN (DBD11NN T)

DBD1101 W field name ALREADY EXISTS AS SPECIFIED IN 'TO' PORTION

Explanation: The field specified is an ID keyword that identifies a particular segment occurrence of subject data. For each subject in the Dictionary, the occurrences of subject data are uniquely identified by this field. The user attempted to change an ID value to a value that has already been stored. This cannot be done.

System Action: Execution continues.

User Response: Verify that the field name was specified correctly.

DBD1102 I DATA HAS BEEN CHANGED FOR subject category subject name

Explanation: Execution of the command has terminated successfully.

User Response: None.

DELETE (DBD12NN T)

DBD1201 I subject category subject name HAS BEEN DELETED

Explanation: Execution of the command has terminated successfully.

User Response: None.

DBD1202 E subject category subject name HAS ALIASES

Explanation: Subject name was not deleted. Since it has aliases, all data for the aliases would be lost if subject name were to be deleted.

System Action: Execution of the command is terminated. Subject is not deleted.

User Response: Subject name can be deleted even though it has aliases by using the UNRESTR operand of the DELETE command.

DBD1203 I subject category subject name DOES NOT EXIST

Explanation: The subject name could not be deleted because it does not exist.

System Action: Execution of the command is terminated. No data in the Dictionary is changed.

User Response: Verify that subject category and subject name were correctly specified.

CHANGE RELATIONSHIP DATA (DBD13NN T)

DBD1301 I DATA HAS BEEN CHANGED UNDER RELATIONSHIP

Explanation: Execution of the command has terminated successfully.

System Action: Execution of the command is terminated. No relationship is established. However, the subject definition is added if it did not exist before the execution of this command.

User Response: None.

DBD1302 E RELATIONSHIP WITH SAME SEQUENCE ATTRIBUTE VALUE ALREADY EXISTS

Explanation: The user attempted to change the sequence field of a relationship, but a relationship with that sequence attribute value already exists.

System Action: Execution of the command is terminated. No data has been changed.

User Response: Specify a different sequence attribute value, or use the DELETE_RELATIONSHIP command to delete the existing relationship. Resubmit the command.

ADD RELATIONSHIP (DBD14NN T)

DBD1401 E NOT DONE, PRIMARY NAMES OF SUBJECTS ARE IDENTICAL

Explanation: Both primary names cannot be the same.

User Response: Select other subject names.

DBD1402 E NOT DONE, CONFLICTING RELATIONSHIP - SUBJECT B CONTAINS SUBJECT A

Explanation: SUBJECT B contains SUBJECT A, so SUBJECT A cannot also contain SUBJECT B.

System Action: Execution of the command is terminated. No relationship is established. However, either or both of the subjects are added if they did not exist before the execution of this command.

User Response: Select other subject names, or use the DELETE_RELATIONSHIP command to delete the conflicting relationship.

DBD1403 E HIERARCHICAL ERROR, NO RELATION BETWEEN A AND C

Explanation: No relationship was established between SUBJECT A and SUBJECT C, because no relationship exists between SUBJECT A and SUBJECT B.

System Action: Execution of the command is terminated. No relationship is established. However, either or both of the subjects are added if they did not exist before the execution of this command.

User Response: Verify also that the subject categories and subject names were specified correctly, or use the ADD_RELATIONSHIP command to establish the relationship.

DBD1404 E HIERARCHICAL ERROR, NO RELATION BETWEEN B AND C

Explanation: No relationship was established between SUBJECT B and SUBJECT C, because no relationship exists between SUBJECT A and SUBJECT B.

System Action: Execution of the command is terminated. No relationship is established. However, either or both of the subjects are added if they did not exist before the execution of this command.

User Response: Verify that the subject categories and subject names were specified correctly, or use the ADD_RELATIONSHIP command to establish the relationship.

DBD1405 I SUBJECT B IS NOW ALIAS OF SUBJECT A

Explanation: The alias relationship was established as specified.

System Action: Execution continues.

User Response: None.

DBD1406 E NOT DONE, BECAUSE OF DATA UNDER SUBJECT B

Explanation: Data is allowed only under the primary name. Since data exists under the secondary name, the relationship was not established.

System Action: Execution of the command is terminated. No relationship is established.

User Response: Use the DELETE command to erase the data. Next, use the ADD command to enter SUBJECT B into the Dictionary again. Then, repeat this command.

DBD1407 E NOT DONE, SUBJECT B IS ALIAS, BUT NOT OF SUBJECT A

Explanation: SUBJECT B can be an alias of only one primary name. Since a conflicting relationship exists, another relationship was not established.

System Action: Execution of the command is terminated. No relationship is established.

User Response: Use the DELETE_RELATIONSHIP command to delete the conflicting relationship, then repeat this command.

DBD1408 E NOT DONE, BECAUSE SUBJECT A IS ALIAS

Explanation: A primary name cannot also be an alias.

System Action: Execution of the command is terminated. No relationship is established.

User Response: Verify that the subject categories and subject names were entered correctly, or use the DELETE_RELATIONSHIP command to delete the conflicting relationship, then

repeat this command.

DBD1409 E NOT DONE, SUBJECT A AND SUBJECT B ARE IDENTICAL

Explanation: The relationship could not be added because subject names must be different.

System Action: The command was not executed.

User Response: Select a different subject name, and resubmit the command.

CHANGE NAME (DBD16NN T)

DBD1601 I NAME HAS BEEN CHANGED

Explanation: The command was executed as requested.

User Response: None.

DBD1602 E NEW RECORD ALREADY EXISTS

Explanation: A name cannot be changed to one that is already stored in the Dictionary.

System Action: Execution of the command is terminated. No change is made to the subject name.

User Response: Select another name or delete the conflicting name and enter the CHANGE_NAME command again to alter the value for field-name.

DBD1603 E OLD AND NEW CATEGORIES NOT IDENTICAL

Explanation: The two names specified in a CHANGE_NAME command must be in the same subject category.

System Action: Execution of the command is terminated.

User Response: Correct either the old subject or the new subject, and resubmit the command.

ADD RELATIONSHIP WITH DATA (DBD17NN T)

DBD1701 I DATA HAS BEEN ADDED TO RELATIONSHIP

Explanation: One or more items were added to the relationship. This message will occur when the segment with element and element contains element relationships are specified.

System Action: Execution of the command has terminated successfully.

User Response: None.

DBD1702 I SEG/DTE LEVEL-66 RELATIONSHIP - START VALUE IS INCORRECT OR NOT SPECIFIED - DEFAULTS TO 32767

Explanation: The user is establishing a segment/element relationship for COBOL level-66 (renames) via the GENFLQ=6 keyword. This message results if the START= keyword was not specified or if the value given was incorrect and informs the user of the default taken.

System Action: The start value is defaulted to 32767.

User Response: None

ADD WITH DATA (DBD18NN T)

DBD1801 I DATA HAS BEEN ADDED TO subject category subject name

Explanation: One or more items were added to this subject name.

System Action: Execution of the command has terminated successfully.

User Response: None.

DELETE DATA (DBD19NN T)

DBD1901 I DATA HAS BEEN DELETED FROM subject-name

Explanation: One or more Dictionary segments were deleted from this subject name.

System Action: Execution of the command has terminated successfully.

User Response: None.

DELETE RELATIONSHIP DATA (DBD20NN T)

DBD2001 I DATA HAS BEEN DELETED UNDER RELATIONSHIP

Explanation: One or more Dictionary segments were deleted.

User Response: None.

DBD2002 W RELATIONSHIP DATA CANNOT BE DELETED WITH THIS COMMAND

Explanation: This command is not currently supported for Extensibility relationships, because the data associated with them has no identifiable groupings.

System Action: Execution of the command is terminated. No data is deleted from the Dictionary.

User Response: The user may delete the desired data with either of two commands. The CHANGE_RELATIONSHIP_DATA command will delete specific attribute data. The DELETE_RELATIONSHIP command will delete the relationship between the two specified subjects and all the relationship data pertaining to that relationship.

COBOL IN AND PLI IN (DBD21NN AND DBD22NN T)

DBD2102 E SOURCE MEMBER membername CANNOT BE FOUND

Explanation: The member name specified on the PLI_IN command could not be located in the specified partitioned data set.

System Action: Execution of other members (if any) continues.

User Response: Check to see that the DD card specified the correct PDS. If so, verify that the PDS contains the referenced member. If not, add the member to PDS.

DBD2103 W NO AVAILABLE OCCURRENCE NUMBER FOR subject name

Explanation: All numbers between global default occurrence and 255 are already in use.

System Action: No entries from this member or sequential data set are added to the Dictionary.

User Response: Correct the name, and respecify it.

DBD2104 E PERMANENT I/O ERROR ON SOURCE INPUT DATASET datasetname

Explanation: An I/O error occurred while attempting to read the named source data set.

System Action: No entries from this member or sequential data set are added to the Dictionary.

DBD2105 S UNABLE TO OPEN SOURCE INPUT DATA SET DDNAME= name

Explanation: The OPEN Macro failed to open the indicated data set. This is usually caused by a missing DD statement.

User Response: Consult the job listing for the cause of the error. Correct the error, and resubmit the command.

System Action: Execution of a COBOL_IN command is terminated.

User Response: Follow normal installation procedures to correct the permanent I/O error.

DBD2106 W COBOL STATEMENT EITHER INVALID OR DOES NOT COMPLY WITH COBOL_IN RESTRICTION (CODE: reasoncode) - statement

Explanation: The indicated COBOL statement "statement" contains a COBOL syntax error, or fails to comply with one of the restrictions of the Dictionary COBOL_IN function. The nature of the error is indicated by "reasoncode."

Code Meaning

LN	The level number contains a nonnumeric character.
NN	A data name is missing from the statement.
RN	The REDEFINES clause in the statement is incomplete.
SE	The statement contains a syntax error, or fails to comply with COBOL IN restrictions.
8N	The data name 1 is missing from this level-88 statement.
8S	The level-88 statement contains a syntax error, or fails to comply with COBOL IN restrictions.
NL	The statement contains a data name longer than 30 characters.

System Action: Syntax check of this member continues, but it is not added to the Dictionary.

User Response: Correct the COBOL input statements, and resubmit the job.

DBD2107 W QUALIFIED NAMES NOT SUPPORTED statement image

Explanation: A qualified name was discovered in the input stream.

System Action: The member or data set will not be added to the Dictionary.

User Response: Remove qualified names from input.

**DBD2108 E FUNCTION NOT AVAILABLE
ONLINE**

Explanation: A PLI_IN command was entered from an online terminal. This function is available only through the batch job stream.

System Action: The command is rejected, and no execution takes place.

User Response: Include the PLI_IN command in a batch job stream.

**DBD2109 E BLOCK SIZE NOT SPECIFIED
FOR PLI_IN DATASET WITH
DDNAME ddname**

Explanation: After opening the input data set DCB, or PLI_IN found that the DCB field DCBBLKSI contains zero.

System Action: PLI_IN execution is terminated.

User Response: Add the block size to the DCB parameter on the DD statement.

**DBD2116 W UNABLE TO OBTAIN
SUFFICIENT STORAGE FOR
INTERNAL CVLTABLE FOR
MEMBER member name**

Explanation: There is insufficient storage in the region to allow this member of the data set to be processed.

System Action: Syntax check continues but this member or sequential data set is not entered in the Dictionary.

User Response: Increase region size or reduce the number of lines of COBOL source statements per member or sequential data set.

**DBD2117 W INVALID OR INCOMPATIBLE
PICTURE, USAGE AND SIGN
CLAUSE FOR ELEMENT element
name**

Explanation: Unable to calculate element length because of invalid or missing clauses in the COBOL source statement.

System Action: Edit continues, but this member or sequential data set is not entered in the Dictionary.

User Response: Correct the COBOL source statement.

**DBD2118 I (membername/PLISEQ) HAS
BEEN ENTERED INTO THE
DICTIONARY**

Explanation: Normal completion response.

System Action: None.

User Response: None.

**DBD2127 I NO INPUT FROM
(membername/PLISEQ)
ENTERED INTO THE
DICTIONARY**

Explanation: A valid data definition was not found in the source input.

System Action: Execution continues with the next member. If none, or input is from a sequential data set, execution of this command terminates.

Note that for PL/I this message only pertains to the structure currently being processed. If the member or sequential data set contained valid structures prior to the current one, they have been entered in the Dictionary.

User Response: Correct the source input.

**DBD2128 W A NON-EMPTY SEGMENT HAS
FILLER AS SEGMENT NAME**

Explanation: A COBOL filler exists at the segment level. The segment level defaults to 01 but may be modified with the SEGLVL= parameter on the COBOL_IN command.

System Action: Execution for edit purposes continues, but this member or sequential data set is not added to the Dictionary.

User Response: Correct COBOL source (include segment name) or specify a smaller SGLVL value.

**DBD2129 W COPY NOT SUPPORTED
STATEMENT statement**

Explanation: The copy function is not supported in COBOL_IN.

System Action: Execution for edit purposes continues, but this member or sequential data set is not added to the Dictionary.

User Response: Correct COBOL source statements.

**DBD2130 W STATEMENT GREATER THAN nn
CHARACTERS _ first card of
statement**

Explanation: A COBOL source statement exceeds the maximum length of nn characters.

System Action: Execution for edit purposes continues, but this member or sequential data set is not added to the Dictionary.

User Response: Revise the COBOL source statement to nn or fewer characters.

DBD2131 T UNABLE TO FREE INTERNAL STORAGE BLOCK(S)

Explanation: The storage module is unable to free some or all of the storage blocks acquired during COBOL_IN or PLI_IN execution.

System Action: Execution is terminated, and the command stream is flushed. Dictionary entries made before the error occurred remain in the Dictionary.

User Response: Notify your IBM representative.

DBD2132 I MORE THAN nn EMBEDDED COMMENTS WITHIN STATEMENT - first statement card

Explanation: More than nn comments are embedded between the start and end of a COBOL record or field statement in the COBOL source.

System Action: The embedded comments above nn are ignored, and execution continues.

User Response: Revise the source statements to limit embedded comments to 100.

DBD2133 W MORE THAN 255 COMMENTS - REMAINDER IGNORED

Explanation: Comments are limited to 255 lines.

System Action: Extra comment lines are ignored and execution continues.

DBD2134 S SYNADAF message

Explanation: The actual message substitutes for "message" above. This message is always preceded by message DBD2104.

System Action: Execution of this member stops. Execution continues with the next member.

User Response: Re-create the COBOL source data set, and rerun the command.

DBD2135 S UNABLE TO OBTAIN BUFFER FOR SOURCE PDS

Explanation: Insufficient storage is available for execution of the source partitioned data set.

System Action: The function is terminated immediately.

User Response: Specify a larger region size for Dictionary execution.

DBD2136 S INSUFFICIENT VIRTUAL STORAGE AVAILABLE:

Explanation: The function indicated in the message cannot be performed because of insufficient storage.

System Action: COBOL_IN is terminated immediately.

User Response: Increase the parameter for region size.

DBD2137 E THE FOLLOWING literal VARIABLE CANNOT BE FOUND: variable

Explanation: The target of a redefining entry or a renaming entry could not be found in the variable-length table.

System Action: All entries for this member are backed out and processing on this member halts.

User Response: Include a COBOL_IN entry for the missing item and resubmit the command.

DBD2138 E CONDITION VARIABLE CANNOT BE FOUND FOR variable

Explanation: DBDCOUT is building the ADD command for a level-88 item, and the condition variable for the named item could not be found in the variable-length table.

System Action: All entries for this member are backed out and processing on this member halts.

User Response: Include a condition variable for the level-88 item and rerun COBOL_IN.

DBD2139 I ALL ENTRIES FROM THIS INPUT MEMBER/SEQUENTIAL DATA SET HAVE BEEN BACKED OUT

Explanation: An error occurred in DBDCOUT which requires a backout of all commands already processed for this member. The backout was successful.

DBD2140 S BACKOUT UNSUCCESSFUL FOR SEGMENT name

Explanation: Self-explanatory.

System Action: The backout halts. Continued execution depends on what the user specified in the FLUSH parameter.

DBD2141 E UNSUCCESSFUL COMPLETION OF THE FOLLOWING GENERATED COMMAND: command

Explanation: Self-explanatory.

System Action: Terminates execution of the command.

User Response: An internal Dictionary problem is indicated. Consult your IBM representative.

DBD2142 W THE CALCULATED LENGTH OF SEGMENT segment name EXCEEDS THE MAXIMUM LENGTH (32767) THAT CAN BE STORED BY THE DICTIONARY

Explanation: The length of a segment is the sum of its subordinate lengths plus any gaps that may occur between items. This segment length total exceeds 32767 bytes and cannot be stored in the Dictionary.

System Action: Execution of this segment is halted. The segment and its subordinates are not entered in the Dictionary. Execution continues with the next segment.

User Response: Correct the structure so that the total length of the segment does not exceed 32767 bytes.

DBD2143 W INVALID SEGMENT segment name IGNORED

Explanation: COBOL_IN analysis has determined that an invalid elementary item at the segment level has been found in the source input, that is, it is an empty segment.

System Action: The statement is ignored, and execution continues with the next acceptable statement.

User Response: Remove the invalid elementary item from segment data definition, or follow the empty segment with at least one element, or remove the empty segment, or verify that the SEGLVL= keyword was specified correctly on the COBOL_IN command.

DBD2145 I NOTHING HAS BEEN ENTERED INTO THE DICTIONARY

Explanation: You entered a COBOL_IN or PLI_IN RECALCULATE_SEGMENT command with UPDATE=NO specified (or defaulted). As a result, no data has been entered into or changed in the Dictionary.

System Action: The appropriate update commands have been generated and the requested reports produced, but the generated commands have not been executed.

User Response: If you wish to have the Dictionary updated as a result of executing the command, specify UPDATE=Y in the command and resubmit it.

DBD2146 I UPDATE DEFAULTED TO NO (UPDATE=N)

Explanation: No UPDATE= parameter was specified in the RECALCULATE_SEGMENT command and the default UPDATE=N was used.

System Action: Execution continues.

User Response: If you wish to have the Dictionary updated with the results of the segment structure recalculation, resubmit the command and specify UPDATE=Y.

DBD2150 I COBOL_IN INPUT LANGUAGE CODE DEFAULTED TO 'C' - (LANG=C)

Explanation: On the COBOL_IN command, the user did not specify a language code under which any entries to the Dictionary would be stored. The Dictionary defaulted this language code to 'C'.

System Action: Execution continues.

User Response: None.

DBD2151 I PLI_IN INPUT LANGUAGE CODE DEFAULTED TO 'B' - (LANG=B)

Explanation: On the PLI_IN command, the user did not specify a language code under which any entries to the Dictionary would be stored. The Dictionary defaulted this language code to 'B'.

System Action: Execution continues.

User Response: None.

DBD2152 W xxxx IS DEFINED AS A BIT ELEMENT; BIT SUPPORT NOT PROVIDED BY COBOL

Explanation: The named element will be skipped in the output COBOL structure and a filler entry created to define the space occupied by the element. The filler is expressed in character format, representing 8-bit multiples.

System Action: Execution continues through the structure.

User Response: None. Warning message only.

DBD2153 I SEGLVL PARM VALUE EXCEEDS COBOL LIMIT - CHANGED TO SEGLVL=49

Explanation: The user has specified a value which would cause the level number of a COBOL record descriptor entry to exceed 49. The value specified was changed to 49.

System Action: Execution continues.

User Response: For COBOL items, the user should specify a segment level value which does not exceed 49.

DBD2200 E A PL/I PICTURE CLAUSE DOES NOT BEGIN WITH A QUOTE

Explanation: The leading character of a PL/I PICTURE clause was not a single quotation mark.

System Action: Execution of the command is terminated.

User Response: Make sure that the PL/I PICTURE clause is a quoted string and that the first character following the equal sign is a single quotation mark.

DBD2201 E WHILE PROCESSING A PL/I PICTURE CLAUSE, THE END OF THE COMMAND WAS REACHED BEFORE THE CLOSING QUOTE WAS FOUND

Explanation: Self-explanatory.

System Action: Execution of the command is terminated.

User Response: Ensure that the PL/I PICTURE is terminated by a single quotation mark.

DBD2202 E AN INVALID CHARACTER HAS BEEN RECOGNIZED IN A PL/I PICTURE CLAUSE

Explanation: An impermissible PL/I PICTURE character has been recognized within the clause. This does not imply full validation of the picture clause, only that an illegal picture character has been found.

System Action: Execution of the command is terminated.

User Response: Make sure that all characters within the quoted PL/I PICTURE string are valid PL/I PICTURE characters.

DBD2203 E A PL/I PICTURE CLAUSE WHICH CONTAINS NO DATA HAS BEEN ENCOUNTERED

Explanation: During execution of a PL/I PICTURE clause, no data was encountered as comprising the clause.

System Action: Execution of the command is terminated.

User Response: Make sure the PICTURE clause contains at least one character of data between the single quotation marks.

DBD2204 E THE LENGTH OF THE PL/I PICTURE CLAUSE EXCEEDS THE DICTIONARY LIMIT OF 30 CHARACTERS

Explanation: The Dictionary will store only 30 characters (including enclosing quotation marks) of data as

a PICTURE clause. The evaluated clause exceeds this limit.

System Action: Execution of this command is terminated.

User Response: Reduce the size of the PICTURE clause or store the PICTURE clause as PLIINFO (PLITYPE=X).

DBD2205 E PL/I INITIAL CALL ATTRIBUTE NOT SUPPORTED BY THE DATA DICTIONARY

Explanation: Self-explanatory.

System Action: Execution of the command is terminated.

User Response: Use a form of the INITIAL attribute supported by the Dictionary or store this clause as PLIINFO (PLITYPE=X).

DBD2206 E A PL/I INITIAL CLAUSE MUST BEGIN WITH A LEFT PARENTHESIS

Explanation: Self-explanatory.

System Action: Execution of this command is terminated.

User Response: Begin the INITIAL clause with a left parenthesis.

DBD2207 E ONE OR BOTH ENCLOSING QUOTES MISSING IN A PL/I INITIAL CLAUSE

Explanation: The PL/I initial clause must be enclosed by quotation marks. Either one or both of these quotation marks is missing from the clause.

System Action: Execution of this command is terminated.

User Response: Correct the initial clause in error, and resubmit the command.

DBD2208 E UNBALANCED PARENTHESES IN A PL/I INITIAL CLAUSE

Explanation: In the PL/I INITIAL clause being evaluated, the number of left parentheses does not match the number of right parentheses.

System Action: Execution of this command is terminated.

User Response: Make sure the numbers of left and right parentheses agree.

DBD2209 E A PL/I INITIAL CLAUSE WHICH CONTAINS NO DATA HAS BEEN ENCOUNTERED

Explanation: During execution of a PL/I INITIAL clause, no data was encountered as comprising the clause.

System Action: Execution of the command is terminated.

User Response: Make sure the INITIAL clause contains at least one character of data between the balanced parentheses.

DBD2210 E THE LENGTH OF THE PL/I INITIAL CLAUSE EXCEEDS THE DICTIONARY LIMIT OF 30 CHARACTERS

Explanation: The Dictionary will store only 30 characters (including enclosing parentheses) of data as an INITIAL clause. The evaluated clause exceeds this limit.

System Action: Execution of this command is terminated.

User Response: Reduce the size of the INITIAL clause, or store the INITIAL clause as PLIINFO (PLIATYPE=X).

DBD2211 E INVALID xxxx VALUE SPECIFIED - REASON CODE=nn

Explanation: The specified value has been found to be invalid for one of the following reasons:

Code	Meaning
04	The value is not numeric.
08	No value was found.
12	The length of the value exceeds the maximum permissible length.
16	The value is outside the allowable range (either too high or too low).
20	Too many dimensions specified in the dimension attribute.
24	A lower bound for a dimension is greater than its upper bound.

System Action: Execution of this command is terminated.

User Response: Correct the value in error, and resubmit the command.

DBD2212 E INVALID DELIMITER IN xxxx CLAUSE

Explanation: In the identified clause, an invalid delimiter was encountered. For example, the length and scale values of a PRECISION attribute may be separated only by a comma. Lower and upper bounds of the dimension clause are delimited by colons; dimensions are separated by commas.

System Action: Execution of this command is terminated.

User Response: Correct the clause in error, and resubmit the command.

DBD2213 E UNBALANCED PARENTHESES ON xxxx CLAUSE

Explanation: On the specified clause, the number of left and right parentheses did not match.

System Action: Execution of this command is terminated.

User Response: Correct the erroneous clause, and resubmit the command.

DBD2214 W CONFLICTING nnnn ATTRIBUTE ENCOUNTERED FOR SEGMENT/ELEMENT nnnn

Explanation: The identified PL/I attribute conflicts with an attribute previously processed for the named item.

System Action: The previous attribute or attribute code is overlaid with the most current.

User Response: Examine the conflicting attribute designations and correct them if the undesired attribute has been stored.

DBD2215 W UNSUPPORTED ATTRIBUTE 'attribute' FOUND, IGNORED.

Explanation: PLI_IN has encountered an attribute not supported by the Dictionary.

System Action: Execution continues.

User Response: If the attribute is at the segment level, the user has three options:

- If the attribute is always to be produced by STRUCTURES_OUT for this segment, then it should be added to PLIDATA for the segment with PLIATYPE=X
- If the attribute is sometimes required for the segment, when the STRUCTURES_OUT command is issued, the SEGATR=attribute parameter should be included with the command
- If the attribute is to be included on all PL/I structures produced by STRUCTURES_OUT, the attribute should be included in the system default module, DBDGFALT.

If the attribute is at the element level, then only option 1 above is available.

DBD2216 E DEFINED ATTRIBUTE WAS FOUND AT OTHER THAN SEGMENT LEVEL

Explanation: The DEFINED attribute is supported only at the segment level.

System Action: Execution of the PLI_IN command is terminated.

User Response: Alter the structure to comply with this restriction, and resubmit the command.

DBD2217 E IMPLIED POSITION SPECIFIED ON STRING OVERLAY DEFINING STATEMENT

Explanation: Overlay defining is not supported in the case in which the base element specified is an element of an array.

System Action: Execution of this command is terminated.

User Response: Alter the structure to comply with this restriction, and resubmit the command.

DBD2218 E BASE ELEMENT IN A DEFINED STATEMENT IS AT SEGMENT LEVEL

Explanation: The base element in a DEFINED statement must be beneath the segment level.

System Action: Execution of the PLI_IN command is terminated.

User Response: Alter the structure to comply with this restriction, and resubmit the command.

DBD2219 E ACCUMULATED LENGTH OF INHERITED DIMENSIONS IS OUT OF RANGE

Explanation: The accumulated length of inherited dimensions for the current element is not in the range 1 to 32767.

System Action: Execution of the PLI_IN command is terminated.

User Response: Alter the structure so that the sum of inherited dimensions is in the range 1 to 32767. Then resubmit the command.

DBD2220 E PL/I STATEMENT EITHER INVALID OR DOES NOT COMPLY WITH PLI_IN RESTRICTION (CODE: XX)

Explanation: The code returned in the message has one of the following meanings:

Code Meaning

AR Dimension value was specified at the segment level.
DN Expected dimension value was not found.
EC End of comment indicator was not found before the end of the member.
EP End of a parenthetical expression was not found before the end of the member.
EQ End of a quoted string was not found before the end of the member.
ID Invalid dimension value was detected:
A lower bound of a dimension is greater than its upper bound; a dimension value is nonnumeric; a dimension value is out of range (-32767,+32767), more than 3 dimension values were specified for an item; more than 15 dimension values were specified for a structure (including inherited dimensions); or the total array size, including inherited dimensions, exceeds 32767.
IL Character or bit length was out of range (00001-32767).
IP Precision length was out of range (001-255).
IS Precision scale was out of range (-128,+127).
LL More than 15 logical levels were specified.
LN Nonnumeric character found instead of level number.
LR Physical level number was out of range (001-255).
NL Length of data name exceeded 31 characters.
NN Name was not found in a factored PL/I attribute clause.
NS Unexpected null statement was found. Statement contained only commas, semicolons, and/or comments.
OP Unbalanced parenthesis present in a parenthetical expression.
OQ Unbalanced quotation marks present in a quoted string.
PC An unidentifiable parenthetical clause was found.
PF Picture specification repetition factor was out of range (00001-32767).
PR End of repetition factor was not found in a picture specification.
PS Picture repetition factor was specified without a picture specification.
QS An unidentifiable quoted string was found.
VI Value for a level number, character or bit length, precision length, precision scale, or repetition factor

VN was not specified.
Nonnumeric character was found in a level number, character or bit length, precision length, precision scale, or repetition factor.
WL Level '1' was expected but not found.

System Action: Execution of the current PLI_IN command is terminated.

User Response: Correct the error, and resubmit the command.

DBD2222 E LENGTH OF ELEMENT OR STRUCTURE IS OUT OF RANGE

Explanation: The length of the current element or the structure is not in the range 1 to 32767.

System Action: Processing of the PLI_IN command is terminated.

User Response: Alter the element or the structure so its length is in the range 1 to 32767. Then resubmit the command.

DBD2223 W END OF INPUT REACHED BEFORE END OF STRUCTURE

Explanation: An end of file was encountered before the end of the structure.

System Action: An attempt is made to process the structure as entered.

User Response: If structure entered is incomplete, you may have to use Dictionary commands to alter it.

GENERAL SERVICE /COMMON FUNCTION (DBD23NN T)

DBD2300 T INSUFFICIENT GETVIS AREA

Explanation: The Dictionary uses the DOS/VS GETVIS macro to dynamically allocate storage. A return code has been received from DOS/VS indicating that sufficient GETVIS area was not available in the partition.

System Action: The task is canceled.

User Response: Change the size parameter on the DOS/VS EXEC statement, or run in a larger partition to give more GETVIS area.

DBD2301 T INVALID FREEVIS PARAMETER

Explanation: The Dictionary uses the DOS/VS FREEVIS macro, and has received a return code from DOS/VS indicating an invalid parameter was passed.

System Action: The task is canceled.

User Response: Obtain a system dump and call your IBM representative.

DBD2304 T AN ERROR WAS FOUND IN A CALL TO THE SYSTEM MESSAGE HANDLER, MSGID=id, MSGRC=nn, NO. OF INSERTS=n, NO. OF INSERT FLAGS=n, DATA LIST AT LOC. hex location

Explanation: A call was made to the Dictionary message handler DBDM. The parameters supplied with the call were incorrect. Either the message number identification could not be found or the number of inserts did not match the requirements of the named message.

System Action: A Dictionary abnormal termination initiated with an identification of 4000.

User Response: If the message number is valid and the inserts match the insert flag count, the appropriate message module may have been omitted from LINKEDIT. If so, include the module in the Dictionary LINKEDIT and rerun the job. If the number is invalid or the inserts do not match the flags, notify your IBM representative.

INTERACTIVE DISPLAY FORMS FACILITY (DBD24NN T TO DBD40NN T)

DBD2400 I PROCESSING COMPLETED SUCCESSFULLY

Explanation: The required processing has been completed and no error conditions have been detected.

System Action: The results of the requested action are shown on the screen.

User Response: Proceed with further Dictionary work.

DBD2410 E IMPROPER OR NO ACTION SELECTED ON FORM

Explanation: An action that is invalid for this processor in the current processing mode (display, edit, or delete) has been specified, or no action was specified prior to pressing the enter key.

System Action: Processing of the form is terminated.

User Response: Enter a valid action.

DBD2411 S SYSTEM ERROR yyyxxxx IN MODULE DBDdyyy

Explanation: A system error has occurred when an attempt was made to retrieve information from the Dictionary. xxxx further defines the error as occurring at a specific

location in module DBDDyyy.

System Action: The definition process is terminated.

User Response: Notify your IBM representative.

DBD2412 E IMPROPER OR NO USER-DATA NUMBER SPECIFIED

Explanation: A number outside the range 1 to 5 has been specified for user data.

System Action: Processing of the form is terminated.

User Response: Correct the number, and resubmit the form.

DBD2413 E IMPROPER OR NO DOWN-COUNT VALUE SPECIFIED

Explanation: The DOWN action was selected and the count supplied was invalid or missing.

System Action: Processing of the form is terminated.

User Response: Reenter the down-count, and resubmit the form.

DBD2414 E INVALID "MODE" CODE SPECIFIED

Explanation: On a PHYSICAL DATABASE STRUCTURE, LOGICAL DATABASE STRUCTURE, SEGMENT-FIELDS, or SUBORDINATE/SUPERIOR FIELDS form, the user has specified a mode code for a segment or field other than 0 or N.

System Action: The form is retransmitted to the user without further processing.

User Response: Correct the erroneous mode code, and reenter the form for processing.

DBD2415 S DL/I ERROR ON DATABASE name STATUS WAS name LAST SEG WAS name

Explanation: A retrieved module has detected a DL/I error during a Dictionary retrieve operation. Information from the PCB is included in the message.

System Action: Processing is terminated.

User Response: Determine the cause of the DL/I error, and correct it.

DBD2416 E RETRIEVAL FAILED: CONCURRENT USER INTERFERENCE SUSPECTED

Explanation: In attempting to perform a REGEN or DOWN retrieval for a form that serves as a window on a long

list, the segment occurrence within the Dictionary data base that is to serve as the key for positioning the retrieval has been found to be missing from the data base. It is most likely that another user has caused this segment to be deleted while the user was working on the window form.

System Action: An empty form is displayed to the user, with this message in the response line.

User Response: Invoke the TOP action, then repeatedly invoke the DOWN action to scan the list to see what changes have been made in it.

DBD2418 E INVALID OR NO UP-COUNT SPECIFIED

Explanation: The UP action was selected on a display form, and the count specified was either invalid or missing.

System Action: Further processing of the display-form is terminated.

User Response: Reenter the up-count, and again select the UP action.

DBD2420 E name IS NOT A VALID HEADER NAME

Explanation: An attempt has been made to enter the interactive display forms facility in an invalid manner or a system error has occurred, where the name of the header form does not match a processor known to the Display Manager.

System Action: The standard HEADER form is invoked to display this message, and the Dictionary transaction is terminated.

User Response: Specify a valid header name, and reenter the request.

DBD2421 S name PROCESSOR NOT KNOWN TO DISPLAY MANAGER

Explanation: A request to call a processor is rejected because the processor is not known to the Display Manager.

System Action: The definition process is terminated.

User Response: Notify your IBM representative.

DBD2422 S requested function INVALID REQUEST IN processor name

Explanation: This processor has requested an invalid action of the Display Manager.

System Action: The definition process is terminated.

User Response: Notify your IBM representative.

DBD2423 S PROCESSOR module name
CANNOT PROCESS FORM form
name

Explanation: A system error has occurred because a form was presented to a processor not able to process such a form.

System Action: The definition process is terminated.

User Response: Notify your IBM representative.

DBD2425 E SUBJECT NAME CHANGE VALID
ONLY WITH REUSE ACTION.

Explanation: The user selected the PROCESS action after changing the subject or data base name on a Segment form or the subject name on a non-Segment form.

System Action: The last subject successfully defined or retrieved using the form is redisplayed to the user. If the form has not yet been used to define a new subject or retrieve a defined subject, the form is redisplayed to the user without alteration.

User Response: Reenter the subject or data base name, if necessary, and select the REUSE action.

DBD2426 E SUBJECT NAME MAY NOT BE
ERASED

Explanation: The user erased (overlaid with minus signs) the subject name and selected the PROCESS or REUSE action.

System Action: The last subject successfully defined or retrieved using the form is redisplayed to the user. If the form has not yet been used to define a new subject or retrieve a defined subject, the form is redisplayed to the user without alteration.

User Response: Enter a subject name, if necessary, and proceed with further Dictionary work.

DBD2450 E LINE NUMBER INVALID -
EITHER UNSPECIFIED OR
ZERO.

Explanation: The line number is required on this form and must be a nonzero number within the acceptable range.

System Action: Processing is terminated.

User Response: Specify a nonzero line number and resubmit the form.

DBD2500 E IN DB NAME AND
RELATIONSHIP MAY NOT BE
SPECIFIED AT THE SAME TIME

Explanation: If a segment in DATABASE: name is specified, the relationship to type field may not be specified.

System Action: Screen processing is terminated.

User Response: Enter either an SEGMENT IN DATABASE name or a relationship request but not both.

DBD2501 E DELETE ACTION INVALID WITH
RELATIONSHIP REQUEST

Explanation: Only the process or display actions are valid with a relationship request.

System Action: Screen processing is terminated.

User Response: Specify either PROCESS or DISPLAY action.

DBD2502 E NO SUBJECT SPECIFIED

Explanation: A subject was not specified on the HEADER form for an action that requires one.

System Action: Screen processing is terminated.

User Response: Reenter input for the header screen.

DBD2505 E DELETE ACTION INVALID WITH
SEGMENT-IN-DATABASE
REQUEST

Explanation: You have specified both a segment name and a SEGMENT IN DATABASE data base name on the HEADER form, and selected the DELETE action. SEGMENT IN DATABASE cannot be specified when the DELETE action is selected.

System Action: HEADER form processing is terminated.

User Response: If you wish to delete the definition for the segment, specify only the segment name on the HEADER form, and select the DELETE action. If you wish to delete the relationship between the segment and the data base, specify both the segment name and the data base name (SEGMENT IN DATABASE) on the HEADER form, select the EDIT action, and then cross out the data base name on the segment form that is displayed.

DBD2506 E INVALID STATUS CODE
status-code SPECIFIED

Explanation: The user specified a status code not in the range A through T or 0 through 9.

System Action: Processing is terminated.

User Response: Specify a valid status code, and resubmit the form.

DBD2507 E INVALID DATA BASE TYPE-CODE n SPECIFIED

Explanation: The user specified a data base type code other than P, L, X, or Y.

System Action: Processing is terminated.

User Response: Specify a valid data base type, and resubmit the form.

DBD2508 E INVALID OCCURRENCE NUMBER

Explanation: An invalid (nonnumeric) character appears in an OCC field.

System Action: The form is redisplayed to the user without further processing.

User Response: Correct the occurrence number, and resubmit the form.

DBD2509 E INVALID LANG (CODE) lang-code SPECIFIED

Explanation: The user specified a language code other than A, B, C, J, K, or L.

System Action: Processing is terminated.

User Response: Specify a valid language code, and resubmit the form.

DBD2510 E INVALID DEFAULT LANG (CODE) VALUE

Explanation: The default qualifier LANG has an invalid character. Valid values are A, B, C, J, K, and L.

System Action: The form is redisplayed to the user without further processing.

User Response: Correct the value, and resubmit the form.

DBD2511 E INVALID DEFAULT OCCURRENCE NUMBER

Explanation: The default qualifier OCC has an invalid (nonnumeric) character, or is greater than 255.

System Action: The form is redisplayed to the user without further processing.

User Response: Correct the value, and resubmit the form.

DBD2512 E INVALID DEFAULT STATUS CODE

Explanation: The default qualifier STAT has an invalid character. Valid values are A through T and 0 through 9.

System Action: The form is redisplayed to the user without further processing.

User Response: Correct the value, and resubmit the form.

DBD2513 E INVALID DEFAULT DATA BASE TYPE-CODE

Explanation: The default qualifier DBTYPE has an invalid character. Valid values are L, P, X, and Y.

System Action: The form is redisplayed to the user without further processing.

User Response: Correct the value, and resubmit the form.

DBD2514 E NO SEGMENT NAME SPECIFIED

Explanation: An IN DB identifier has been received without any segment identifier.

System Action: The form is redisplayed to the user without further processing.

User Response: Specify a segment name, or choose another subject.

DBD2515 E RELATIONSHIP ERROR

Explanation: The subject code specified in the (RELATIONSHIP) field is either not a valid code, or it specifies a related subject category that is treated by one of the special relationship forms.

System Action: The form is redisplayed to the user without further processing.

User Response: Specify a valid related subject category code or none.

DBD2520 E CATEGORY MUST BE SPECIFIED

Explanation: The subject category was not specified on the HEADER form.

System Action: Processing is terminated.

User Response: Specify a category name, or number and resubmit the form.

DBD2521 E INVALID CATEGORY NUMBER SPECIFIED

Explanation: An invalid category number was specified on the HEADER form. The valid range of category numbers is 1 through 255.

System Action: Processing is terminated.

User Response: Specify a valid category number, and resubmit the form.

DBD2522 E INTERNAL ERROR HAS OCCURRED IN HEADER PROCESSOR

Explanation: An unexpected return code was encountered during Header processing.

System Action: Processing is terminated.

User Response: Notify your IBM representative.

DBD2523 E CODE FOR A SUBJECT IN AN EXT CATEGORY MUST BE BLANK

Explanation: A nonblank subject code was specified for a subject in an Extensibility category.

System Action: Processing is terminated.

User Response: Blank the subject code, and resubmit the form.

DBD2524 E INVALID SUBJECT CODE SPECIFIED FOR THIS CATEGORY

Explanation: An invalid subject code was specified for a subject in an "old" category.

System Action: Processing is terminated.

User Response: Correct the subject code, and resubmit the form.

DBD2525 E INVALID OCCURRENCE SPECIFIED FOR THIS CATEGORY

Explanation: An invalid occurrence was specified for a subject in an "old" category.

System Action: Processing is terminated.

User Response: Correct the occurrence, and resubmit the form.

DBD2526 E INVALID RELATIONSHIP REQUEST

Explanation: An attempt has been made to display relationships between a subject in an extensibility category and "xSY0" (all system objects). This is an invalid request.

System Action: No further processing is performed.

User Response: You may display the relationships between a subject in an extensibility category and one system category or the relationships between a subject in an "old" category and "xSY0" (all system objects). Change either the CATEGORY NBR/NAME field or the -RELATIONSHIPS TO field accordingly and resubmit the form.

DBD2530 E CATEGORY category-name NOT INSTALLED

Explanation: The category specified in category-name has not been installed.

System Action: No further processing is performed.

User Response: Request that the category be installed. When this has been done, resubmit the form.

DBD2599 I DB/DC DICTIONARY FACILITY PROCESSING COMPLETE

Explanation: The QUIT action has been detected on the HEADER form.

System Action: Processing of the interactive display forms facility is terminated.

User Response: If further processing is required, a new /FORMAT HDRMOD or /FORMAT DISPMOD must be entered.

DBD2600 I SUBJECT DATABASE NOT DEFINED IN DICTIONARY

Explanation: No definition exists in the Dictionary for a data base having the name, status code, and (implied) data base type specified by the user on the HEADER form.

System Action: The data base identifier is erased from the reference area, and the attribute bytes are set in the I/O area so that the identifier will be displayed "protected" and in lowlight.

User Response: An informational message. In display or delete modes, select the HDR action to return to the HEADER form.

DBD2602 E SUBJECT DATABASE DEFINITION DOES NOT EXIST

Explanation: The user has attempted to delete the definition for a data base when no definition in fact exists. Alternatively, the user has accessed a second-level data base definition form when the subject data base is undefined.

System Action: The form is redisplayed to the user without further processing.

User Response: Select the HDR action to return to the HEADER form.

DBD2604 E DSG-NUMBER MAY NOT BE MODIFIED

Explanation: The user has attempted to change (not erase) one of the DSG-numbers displayed in the table of data set-group specifications at the bottom of the PHYSICAL DATABASE form.

System Action: The DSG-number is restored. The form is redisplayed to the user without further processing.

User Response: The DSG-number associated with a data set-group specification cannot be changed directly on the PHYSICAL DATABASE form. The user can, however, copy the data set-group specification into the line in the table having the desired DSG-number, and delete the old specification by crossing out the DSG-number associated with that specification.

DBD2605 E "INSERT/MOVE" OR "BEFORE" SEGMENT ID MISSING

Explanation: The user has selected either the INSERT or MOVE action on the PHYSICAL or LOGICAL DATABASE STRUCTURE form, and has failed to specify either the identifier for the segment to be inserted (or moved), or the identifier for the segment before which that segment is to be inserted (or moved).

System Action: The form is redisplayed to the user without further processing.

User Response: Enter the missing identifier into the form, and reenter the form, selecting either the INSERT or the MOVE action again.

DBD2606 E SEGMENT TO BE RELATED ALREADY DEFINED

Explanation: The user has attempted to relate (and implicitly define) a segment to the subject data base (either by entering the segment ID into the table, or by using the INSERT action) with MODE= N, and a definition for a segment having that identifier already exists within the Dictionary.

System Action: The form is redisplayed to the user without further processing; the unprocessed segment ID is displayed in lowlight.

User Response: Either set MODE= 0 for the segment, or change the segment ID.

DBD2607 E SEGMENT TO BE RELATED IS NOT DEFINED

Explanation: The user has attempted to relate a segment to the subject data base (either by entering the segment ID into the table, or by using the INSERT action) with MODE= 0, and no definition exists within the Dictionary for a segment having the specified ID.

System Action: The form is retransmitted to the user without further processing. The unprocessed segment ID is displayed in lowlight.

User Response: Either set MODE= N for the segment, or correct the segment ID.

DBD2608 E 'INSERT' SEGMENT IDENTIFIER MAY NOT BE ERASED

Explanation: After it has been processed, the user has attempted to modify the identifier for the INSERT segment in the INSERT/MOVE line of the form.

System Action: The originally specified segment ID is restored, and the form redisplayed to the user without further processing.

User Response: Complete processing of the INSERT/MOVE segment specification before trying to make other corrections in the form.

DBD2609 E ROLE-CODE INCONSISTENT WITH DATABASE TYPE

Explanation: On the PHYSICAL DATABASE STRUCTURE form, the user has specified a role-code value other than P or V for a segment. (P and V are the only valid roles for segments related to a physical data base.)

System Action: The form is redisplayed to the user without further processing. The unprocessed role-code value is displayed in lowlight.

User Response: Correct the erroneous role-code value, and reenter the form for execution.

DBD2610 E SEGMENT IDENTIFIER MAY NOT BE MODIFIED

Explanation: The user has attempted to modify (not erase) a segment identifier in the table comprising the PHYSICAL or LOGICAL DATABASE STRUCTURE form.

System Action: The previous segment identifier is restored in the form, and the form is redisplayed to the user without further processing.

User Response: Change the name of the segment on the segment form displaying the segment definition.

**DBD2611 E ROLE-CODE MAY NOT BE
MODIFIED**

Explanation: The user has attempted to change the role-code specified for one of the segment entries displayed in the table on the PHYSICAL DATABASE STRUCTURE form.

System Action: The previously specified role-code is restored in the form, and the form is redisplayed to the user without further processing.

**DBD2613 E SEC-INDEX SEGMENT NAME
MUST BE SPECIFIED**

Explanation: On the SECONDARY INDEX DATABASE or SECONDARY INDEX form, the user has selected the 5-INDEX action, and failed to specify a SEC INDEX (segment) NAME.

System Action: The form is redisplayed to the user without further processing.

User Response: Enter a segment ID into the SEC INDEX NAME fields before selecting the INDEX action.

**DBD2614 I INDEX POINTER SEGMENT
DEFINITION DELETED**

Explanation: The user has erased the INDEX-SEG segment name on the PRIMARY INDEX DATABASE form; the definition for that pointer segment has been deleted from the Dictionary.

System Action: All fields on the form containing definitional data dependent upon the definition of the index pointer segment are cleared. The form is then redisplayed to the user without further processing.

User Response: An informational message. You may specify a new index pointer segment on the form.

**DBD2615 E INDEX POINTER SEGMENT NOT
SPECIFIED**

Explanation: On a PRIMARY INDEX DATABASE or SECONDARY INDEX form, the user has attempted to enter definitional data dependent upon the specification of the INDEX-SEG pointer segment without specifying a pointer segment identifier.

System Action: The form is redisplayed to the user without further processing. The unprocessed specifications are displayed in lowlight.

User Response: Enter an identifier for the INDEX-SEG pointer segment and reenter the form for processing, or select the REGEN action to have the dependent specifications erased from the form.

DBD2616 E KEY-FIELD NOT SPECIFIED

Explanation: On a PRIMARY INDEX DATABASE or SECONDARY INDEX form, the user has entered specifications dependent upon the specification of the KEY-FIELD field without specifying an identifier for the key field.

System Action: The form is redisplayed to the user without further processing. The unprocessed specifications are displayed in lowlight.

User Response: Enter an identifier for the KEY-FIELD field, and reenter the form for processing; alternatively, select the REGEN action to erase the dependent specifications from the form.

**DBD2617 E TARGET SEGMENT NOT
SPECIFIED**

Explanation: On a PRIMARY INDEX DATABASE or SECONDARY INDEX form, the user has attempted to enter specifications of the TARGET-SEG segment identifier without specifying an ID for the target segment.

System Action: The form is redisplayed to the user without further processing. The unprocessed specifications are displayed in lowlight.

User Response: Either enter a TARGET-SEG segment identifier and reenter the form for processing, or select the REGEN action to erase the dependent specifications from the form.

**DBD2618 E INDEX DATABASE DEFINITION
DOES NOT EXIST**

Explanation: The user has accessed the SECONDARY INDEX form, and the subject index data base is not defined in the Dictionary.

System Action: The form is redisplayed to the user without further processing.

User Response: Select the RTN action to return to the SECONDARY INDEX DATABASE form in order to establish a definition for the index data base.

**DBD2619 I SECONDARY INDEX NOT
DEFINED**

Explanation: The user has accessed a SECONDARY INDEX form using the INDEX action on either the SECONDARY INDEX DATABASE or another SECONDARY INDEX form, and no definition for the secondary index pointer segment having the name specified by the user was found in the Dictionary.

System Action: The INDEX-SEG segment identifier is erased from the reference area, and the attribute bits set in the I/O area so that the identifier will be displayed in lowlight.

User Response: An informational message. If you wish to have the index pointer segment defined, submit the form for processing at least once.

DBD2620 E INDEX DEFINITION DOES NOT EXIST

Explanation: The user has attempted to delete the definition for a secondary index when no definition for that index (actually, the index pointer segment) does in fact exist within the Dictionary.

System Action: The form is redisplayed to the user without further processing.

User Response: Select the RTN action to cause the SECONDARY INDEX DATABASE form to be displayed; alternatively, select the HDR action to return to the header form.

DBD2621 E ONE OR MORE SEGMENTS DO NOT HAVE 'DL/I' ALIASES

Explanation: In attempting to display the specified data base structure form, one or more of the segments that would appear as entries within the window of the form have been found not to have DL/I (CODE=A) aliases.

System Action: The truncated primary name of each such segment is displayed on the form in lowlight. The user is automatically forced into display mode and cannot make changes in the contents of the form.

User Response: Return to the HEADER form, then access the appropriate segment and subordinate ALIAS form to assign DL/I aliases to each of the segments at fault.

DBD2622 E (KEY-FIELD) HAS NO 'DL/I' ALIAS

Explanation: When retrieving information relevant to the KEY-FIELD field within the primary or secondary index pointer segment, the key field was found not to have a DL/I alias.

System Action: The form is displayed to the user with the KEY-FIELD identifier in lowlight. The user is forced into display mode to prevent processing any modifications in the contents of the form.

User Response: The FIELD and subordinate ALIAS forms should be accessed to assign a DL/I alias to the KEY-FIELD field.

DBD2623 E (INDEX-SEG) HAS NO "DL/I" ALIAS

Explanation: When retrieving information relevant to the INDEX-SEG segment for the primary index data base, the index pointer segment has been found to have no DL/I alias.

System Action: The form is displayed to the user with the truncated primary name for the INDEX-SEG segment in lowlight. The user is automatically forced into display mode to prevent processing of any modifications in the contents of the form.

User Response: Access the segment and subordinate aliases form to assign an alias to the INDEX-SEG segment.

DBD2624 E (INDEX-SEG) IDENTIFIER MAY NOT BE CHANGED

Explanation: The user has attempted to change (not erase) the INDEX-SEG segment identifier displayed on the form.

System Action: The original identifier is restored to the form, and the form is retransmitted to the user.

User Response: If you wish to change the name of the INDEX-SEG segment, use the command language.

DBD2625 E (KEY-FIELD) IDENTIFIER MAY NOT BE CHANGED

Explanation: The user has attempted to change, not erase, the KEY-FIELD field identifier displayed on the form.

System Action: The original identifier is restored to the form, and the form is retransmitted to the user.

User Response: If you wish to change the name of the KEY-FIELD field, use the command language.

DBD2626 E DATA MAY NOT BE ENTERED WITHOUT RELATED SEGMENT IDENTIFIER

Explanation: The user has entered relationship data into a row in the table containing no segment identifier.

System Action: The form is retransmitted to the user without further processing.

User Response: Either enter a segment identifier to go with the data, or erase the offending entries from the form.

DBD2627 E (KEY FIELD) 'START=' VALUE MAY NOT BE ERASED

Explanation: The user has attempted to erase the START= value for the key field. This value serves as the key in the relationship between the key field and the index pointer segment and cannot be deleted.

System Action: The value previously specified is restored to the form and the form is retransmitted to the user without further processing.

User Response: Since the START= value is a required parameter on every segment-element relationship, no alternative user response is possible.

DBD2628 E (KEY-FIELD) 'SEQ=' VALUE MAY NOT BE ERASED

Explanation: The user has attempted to erase the SEQ= value for the key field. This value serves to determine which of the fields related to the index pointer segment is the key field, and it cannot be erased.

System Action: The value is restored to the form, and the form is retransmitted to the user without further processing.

User Response: Define the segment on the appropriate segment screen.

DBD2629 E SPECIFIED SEGMENT NAME IS NOT A 'DL/I' ALIAS

Explanation: The specified name does not have a code of 'A', which is required for segments in a data base structure.

System Action: Processing of the current form is terminated.

User Response: Specify a DL/I alias, and resubmit the form.

DBD2630 I REGEN FORCED BECAUSE OF CHANGE TO DBACCS VALUE

Explanation: The user has entered or changed the DBACCS value in such a way that the values displayed in the DD2/OVFLW column must be reinterpreted. The form has been automatically regenerated so as to display the correct values in the DD2/OVFLW column.

System Action: As indicated above, the form has been automatically regenerated before being displayed to the user.

User Response: If any entries into the form were lost because of the regeneration, they must be reentered and processed.

DBD3000 E INDEX-SEGMENT NOT DEFINED IN DICTIONARY

Explanation: The user has accessed the XDFLD form, and no definition for the INDEX-SEG segment exists within the Dictionary.

System Action: The form is displayed to the user without further processing.

User Response: Select either the RTN or HDR action, and reenter the form.

DBD3001 E INDEX DATABASE NOT DEFINED IN DICTIONARY

Explanation: The user has accessed the XDFLD form, and no definition for the index data base has been found in the Dictionary.

System Action: The form is redisplayed to the user without further processing.

User Response: Select either the RTN or HDR action, and reenter the form.

DBD3002 E TARGET SEGMENT NOT SPECIFIED

Explanation: On the XDFLD form, the user has attempted to enter specifications dependent upon the TARGET-SEG segment, and has failed to specify an identifier for the segment itself.

System Action: The form is redisplayed to the user without further processing. The unprocessed specifications are displayed in lowlight.

User Response: Enter a TARGET-SEG segment identifier, and reenter the form for processing; alternatively, select the REGEN action to erase the dependent specifications from the form.

DBD3100 I SUBJECT SEGMENT NOT DEFINED IN DICTIONARY

Explanation: No definition has been found in the Dictionary for a segment having the identifier specified by the header form.

System Action: The segment identifier is erased from the reference area, and the attribute bytes are set in the I/O area so that the identifier will be displayed in lowlight and protected.

User Response: If you wish to establish a definition for a segment having the specified identifier, submit the form for processing at least once.

DBD3102 E SUBJECT SEGMENT DEFINITION DOES NOT EXIST

Explanation: The user has attempted to delete the definition for a segment when no definition for that segment in fact exists in the Dictionary.

System Action: The form is redisplayed to the user without further processing.

User Response: Select the HDR action to return to the header form.

DBD3104 E DATABASE IDENTIFIER MAY NOT BE CHANGED

Explanation: The user has attempted to change (not erase) the name of the related data base on a segment form.

System Action: The data base identifier is restored in the form, and the form is redisplayed to the user without further processing.

User Response: If you wish to change the name of the data base, do so on the appropriate data base definition form.

DBD3105 E SEGMENT-DATABASE RELATIONSHIP DOES NOT EXIST

Explanation: On the no-role SEGMENT form, the user has attempted to specify a role-code value before specifying the identifier for the related data base.

System Action: The form is redisplayed to the user without further processing. The unprocessed role-code value is displayed in lowlight.

User Response: Enter the identifier for the related data base and reenter the form for processing, or select the REGEN action to erase the unprocessed role-code value.

DBD3106 E ROLE name INCONSISTENT WITH DATABASE CODE code

Explanation: On the no-role SEGMENT form, the user has specified a role-code value that is inconsistent with the data base type code of the related data base.

System Action: The form is redisplayed to the user without further processing. The unprocessed role-code is displayed in lowlight.

User Response: Correct the specified role-code value and reenter the form for processing, or select the REGEN action to erase the erroneous specification.

DBD3110 E LCHILD SEGMENT NAME NOT SPECIFIED

Explanation: The LCHILD action was requested, but no LCHILD segment name was specified.

System Action: Processing is discontinued for this entry.

User Response: Enter an LCHILD segment name.

DBD3111 E PARENT NAME MUST BE CODE 'A' FOR LCHILD REQUEST

Explanation: The parent segment must be specified by its assembler (code=A) name, because the segment names in parent-child relationships are limited to eight characters.

System Action: Processing is discontinued for this input.

User Response: Go back to the HEADER form and access this physical segment definition by its a (assembler) name. If it has no assembler name, an alias with code A must first be defined.

DBD3120 E KEY-FIELD IDENTIFIER MAY NOT BE CHANGED

Explanation: The user has attempted to change the key-field name on the PRIMARY INDEX DATABASE or SECONDARY INDEX form.

System Action: Processing is terminated, and the form is retransmitted to the user.

User Response: To change the key field, first delete the old name and then add the new key-field name.

DBD3130 I LCHILD RELATIONSHIP IS NOT DEFINED

Explanation: The specified logical child segment relationship is not defined for the specified segment/data base combination.

System Action: The specified LCHILD segment name is displayed on the LCHILD form in lowlight and protected.

User Response: Enter PROC to establish the relationship.

DBD3131 E LCHILD SEGMENT NAME MAY NOT BE ERASED

Explanation: The user has erased the logical child name on the segment logical-child form.

System Action: Processing is terminated and the LCHILD form is retransmitted to the user.

User Response: Specify action 3, DELETE, to delete the relationship between the logical child and its parent.

DBD3141 E THE KEYFIELD IS NOT SPECIFIED

Explanation: The user has attempted to add key field attributes, but the key field itself is not specified.

System Action: No change is made in the Dictionary for the key field or key field attributes.

User Response: Specify the key field.

DBD3150 W SUBJECT SEGMENT IDENTIFIER IS NOT 'DL/I' ALIAS

Explanation: The user has attempted to access the XDFLD form from a secondary index pointer segment form displaying a non-DL/I alias for the subject segment. Since only eight characters of the segment name are displayed on the XDFLD form, processing errors could occur.

System Action: The XDFLD action is not executed. The form is retransmitted to the user.

User Response: If you wish to enter XDFLD specifications for a segment, access the segment definition using its DL/I alias.

DBD3175 I FIELD NOT DEFINED, OTHER PARAMETERS ARE IGNORED

Explanation: The name of the field to be related to the subject segment has not been specified, but other parameters have been detected.

System Action: Processing of the current form is terminated.

User Response: Specify a field name, and resubmit the form.

DBD3176 E RELATED FIELD IDENTIFIER MAY NOT BE MODIFIED

Explanation: The user has attempted to modify a field identifier on the segment field form. A field identifier may be either added or deleted.

System Action: Form processing is terminated.

User Response: You can modify the field identifier via appropriate DELETE and ADD action.

DBD3177 E FIELD TO BE RELATED ALREADY DEFINED

Explanation: The user has attempted to relate a field to a subject segment that is already defined in the Dictionary, but the mode specified or implied was "new."

System Action: The form is redisplayed to the user without further processing. The unprocessed field

identifier is displayed in lowlight.

User Response: Specify a mode of 0 (old).

DBD3178 E FIELD TO BE RELATED IS NOT DEFINED

Explanation: The user has attempted to relate a field to a subject segment that is not defined in the Dictionary, but the mode specified was "old."

System Action: The form is redisplayed to the user without further processing. The unprocessed field identifier is displayed in lowlight.

User Response: Specify a mode of N (new).

DBD3200 I SUBJECT FIELD NOT DEFINED IN DICTIONARY

Explanation: The subject field identifier is not defined in the Dictionary.

System Action: The form is presented to the user for further processing.

User Response: Select the PROC action to cause an entry for the subject field to be placed in the Dictionary.

DBD3206 E IMPROPER OR NO 'RELATED-FIELDS' CODE SPECIFIED

Explanation: The user has specified an invalid related-fields code. Valid specifications are SUB and SUP.

System Action: The form is presented to the user without further processing.

User Response: Specify SUB or SUP, or select another action.

DBD3208 E FIELD NOT DEFINED -- CANNOT BE DELETED

Explanation: An attempt to delete a subject field identifier has been detected, but this identifier does not exist in the Dictionary.

System Action: The form is presented to the user without further processing, with the subject name restored.

User Response: Select another action.

DBD3220 E SUBJECT FIELD DEFINITION DOES NOT EXIST

Explanation: The subject field identifier does not exist in the Dictionary. Further processing cannot be done until a definition exists.

System Action: The form is presented to the user without further processing.

User Response: Return to the field definition form, and define the field.

DBD3226 I SEGMENT NOT DEFINED, OTHER PARAMETERS ARE IGNORED

Explanation: The name of the segment to be related to the subject field has not been specified, but other parameters have been detected.

System Action: Processing of the current form is terminated.

User Response: Specify a segment name, and resubmit the form.

DBD3227 E SEGMENT IDENTIFIER MAY NOT BE MODIFIED

Explanation: The user has attempted to modify a segment identifier on the FIELD SEGMENTS form. A segment identifier may only be added or deleted.

System Action: Form processing is terminated.

User Response: The user can modify a segment identifier via appropriate DELETE and ADD actions.

DBD3228 E SEGMENT TO BE RELATED IS NOT DEFINED

Explanation: A segment identifier has been added to the list of segments on the FIELD SEGMENTS form, but that segment does not exist in the data base.

System Action: Processing is terminated and the completed operations (if any) are displayed to the user.

User Response: If the user wishes to establish a relationship to a segment subject having the specified name, he must first create a Dictionary entry for that segment using either the ADD or the ADD_RELATIONSHIP command, or the SEGMENT display form.

DBD3229 E START VALUE MAY NOT BE ERASED

Explanation: The user has attempted to erase the value under START for some line on the screen.

System Action: The previously specified value is restored, and processing is terminated.

User Response: You must define START for all relationships between segments and fields.

DBD3245 I FIELD NOT DEFINED, OTHER PARAMETERS ARE IGNORED

Explanation: The name of the field to be related to the subject field has not been specified, but other parameters have been detected.

System Action: Processing of the current form is terminated.

User Response: Specify a field name, and resubmit the form.

DBD3246 E RELATED FIELD IDENTIFIER MAY NOT BE MODIFIED

Explanation: The user has attempted to modify a field identifier on the SUBORDINATE/SUPERIOR FIELDS form. A field identifier may only be added or deleted.

System Action: Form processing is terminated.

User Response: The user can modify the field identifier via appropriate DELETE and ADD actions.

DBD3248 E FIELD TO BE RELATED ALREADY DEFINED

Explanation: The user has attempted to relate a field to a subject field that is already defined in the Dictionary, but the mode specified or implied was "new."

System Action: The form is presented to the user without further processing. The unprocessed field identifier is displayed in lowlight.

User Response: Specify a mode of 0 (old).

DBD3249 E FIELD TO BE RELATED IS NOT DEFINED

Explanation: The field to be related to the subject field is not yet defined in the Dictionary, but the mode specified was "new."

System Action: The form is redisplayed to the user without further processing. The unprocessed field identifier is displayed in lowlight.

User Response: Specify a mode of N (new).

DBD3300 I SUBJECT name NOT DEFINED IN DICTIONARY

Explanation: There exists no entry in the Dictionary for the specified subject; further specification will cause the subject to be created.

System Action: The presented form assumes that a new definition is desired.

User Response: Return to the header form and respecify the desired subject, or continue with a new definition.

DBD3301 E name TO BE DELETED IS NOT DEFINED

Explanation: An attempt to delete an entry that does not exist has been made.

System Action: The form is redisplayed without further processing.

User Response: Select the HDR action to return to the HEADER form.

DBD3310 E SUBJECT SYSDEF DEFINITION DOES NOT EXIST

Explanation: You have accessed a second-level SYSDEF form where the subject SYSDEF is undefined.

System Action: The form is redisplayed without further processing.

User Response: Select the HDR action to return to the HEADER form.

DBD3311 E name TO BE RELATED IS NOT DEFINED

Explanation: A PSB, transaction, or data base name that does not exist in the Dictionary has been detected.

System Action: Processing of the form is terminated.

User Response: Correct the form and resubmit it, or, if the PSB, transaction, or data base name is correct, return to the HEADER form and define the subject and reprocess the form.

DBD3312 E item IDENTIFIER MAY NOT BE MODIFIED

Explanation: An attempt to modify the PSB, transaction, or data base identifier has been detected.

System Action: Processing of the form is terminated.

User Response: You can modify the name with appropriate DELETE and ADD actions.

DBD3313 I item NOT DEFINED, OTHER PARAMETERS IGNORED

Explanation: The PSB, transaction, or data base has not been specified but other parameters have been detected.

System Action: Processing of the form is terminated.

User Response: Supply a subject name and reenter.

DBD3350 E SUBJECT PSB DEFINITION DOES NOT EXIST

Explanation: You have accessed the PSB-TRANSACTION definition when the subject PSB is not defined.

System Action: The form is redisplayed without further processing.

User Response: Select the HDR action, and return to the HEADER form.

DBD3351 E TRAN TO BE RELATED IS NOT DEFINED

Explanation: The TRANSACTION name entered on the PSB-TRANSACTION form does not exist in the Dictionary.

System Action: Processing of the form is terminated.

User Response: Correct the form and resubmit it, or, if the TRANSACTION name is correct, return to the HEADER form, define it, and reprocess the form.

DBD3352 E TRAN IDENTIFIER MAY NOT BE MODIFIED

Explanation: You have attempted to modify the TRANSACTION identifier.

System Action: Processing of the form is terminated.

User Response: You can modify the name with the appropriate DELETE and ADD actions.

DBD3353 I TRAN NOT DEFINED, OTHER PARAMETERS IGNORED

Explanation: The TRANSACTION has not been specified, but other parameters have been detected.

System Action: Processing of the form is terminated.

User Response: Supply a subject name, and reenter the transaction.

DBD3400 I SUBJECT PCB NOT DEFINED IN DICTIONARY

Explanation: There exists no entry in the Dictionary for the specified subject; further specification will cause the subject to be created.

System Action: The presented form assumes that a new definition is desired.

User Response: Return to the HEADER form and respecify the desired subject, or continue with a new definition.

DBD3401 E SUBJECT PCB DEFINITION DOES NOT EXIST

Explanation: An attempt to delete an entry from an edit-mode form has been detected.

System Action: Processing of the form is terminated.

User Response: Correct the form and resubmit it, or return to the HEADER form and change processing modes.

DBD3402 E CONFLICTING "TYPE" AND ATTRIBUTES SPECIFIED

Explanation: Values have been supplied for parameters which pertain to a type different from the one specified.

System Action: Processing is terminated, and the form is retransmitted to the user.

User Response: Erase the parameter values that conflict, or delete the PCB and define a new type.

DBD3403 E DATABASE IDENTIFIER MAY NOT BE MODIFIED

Explanation: An attempt to modify one of the components of the data base identifier has been made.

System Action: Processing of the form is terminated.

User Response: If it is necessary to change the identifier, delete the current one and create the new one.

DBD3404 E PCB TYPE MAY NOT BE CHANGED

Explanation: An attempt to change the type field on the form has been made.

System Action: Processing of the form is terminated.

User Response: If it is necessary to change the type, delete and re-create the PCB entry.

DBD3405 E PCB TYPE INVALID -- MUST BE D, T, OR G

Explanation: The value specified for type is not D, T, or G.

System Action: Processing of the form is terminated.

User Response: Correct the form, and resubmit the form.

DBD3406 E SUBJECT PCB IS NOT A DATABASE-PCB

Explanation: The DB-SENSEGS action was selected, but the subject PCB is not a type D PCB.

System Action: Processing of the form is terminated.

User Response: Select another action.

DBD3407 E SENSEG IDENTIFIER MAY NOT BE MODIFIED

Explanation: An attempt to modify the SENSEG identifier has been detected.

System Action: Processing of the form is terminated.

User Response: You can modify a SENSEG name with appropriate DELETE and ADD actions.

DBD3408 E SENSEG TO BE RELATED IS NOT DEFINED

Explanation: A SENSEG name that does not exist in the Dictionary has been detected.

System Action: Processing of the form is terminated.

User Response: Correct the form and resubmit it, or, if the SENSEG name is correct, return to the HEADER form, define the SENSEG, and reprocess the form.

DBD3409 E NO, OR MORE THAN ONE SENSEG SELECTED FOR 'MORE INDICES' ACTION

Explanation: The MORE-INDICES action was selected and either no or multiple question marks (?) have been detected. Processing of the form is terminated.

User Response: Correct the form and resubmit it, or select another action.

DBD3410 E 'INSERT/MOVE' OR 'BEFORE' SENSEG IS MISSING

Explanation: The INSERT or MOVE action was selected and either the name of the SENSEG or the name of the SENSEG after which the SENSEG is to be inserted or moved is missing.

System Action: Processing of the form is terminated.

User Response: Correct the form and resubmit it, or select another action.

DBD3411 I SENSEG NOT DEFINED, OTHER PARAMETERS ARE IGNORED

Explanation: The SENSEG has not been specified, but other parameters have been detected.

System Action: Processing of the current form is terminated.

User Response: Specify a SENSEG name, and reenter the form.

DBD3412 E SUBJECT SENSEG NOT DEFINED IN THE DICTIONARY

Explanation: There exists no entry in the Dictionary for the specified SENSEG.

System Action: Processing of the current form is terminated.

User Response: Return to the HEADER form, define the SENSEG and reprocess the form.

DBD3413 E SPECIFIED SENSEG NAME IS NOT A DL/I ALIAS

Explanation: The specified name does not have a code of A, which is required for all sensitive segments.

System Action: Processing of the current form is terminated.

User Response: Specify a DL/I alias, and resubmit the job.

DBD3414 E 'INSERT' SEGMENT IDENTIFIER MAY NOT BE ERASED

Explanation: After it has been processed, the user has attempted to modify the identifier for the INSERT segment in the INSERT/MOVE line of the form.

System Action: The originally specified segment ID is restored, and the form redisplayed to the user without further processing.

User Response: Complete processing of the INSERT/MOVE segment specification before trying to make other corrections in the form.

DBD3420 E ONE OR MORE SENSEGS DO NOT HAVE 'DL/I' ALIASES

Explanation: In attempting to display the PCB SENSEG form, one or more of the segments that would appear as entries within the window of the form have been found not to have DL/I (CODE=A) aliases.

System Action: The truncated primary name of each such segment is displayed on the form in lowlight. The user is automatically forced into display mode, so that processing of any changes in the contents of the form is prohibited.

User Response: Return to the HEADER form, then access the appropriate segment and subordinate ALIAS form to assign DL/I aliases to each of the segments.

DBD3900 E LINE NUMBER MAY NOT BE MODIFIED

Explanation: The user has attempted to modify an existing line number. The interactive display forms facility does not support modification.

System Action: The current line number associated with the entry is restored, and the form is redisplayed without further processing.

User Response: You can modify the line numbers by appropriate DELETE and ADD actions.

DBD3901 E SUBJECT ENTITY NOT DEFINED IN DICTIONARY

Explanation: The subject entity is not yet defined in the Dictionary. Descriptive and/or relational information entered cannot be processed.

System Action: The display form is redisplayed and the form is redisplayed without further processing.

User Response: Provide subject entity definitions prior to entry of descriptive and/or relational data.

DBD3902 E ALIAS IDENTIFIER MAY NOT BE MODIFIED

Explanation: The user has attempted to modify an existing alias name. The interactive display forms facility does not support this modification.

System Action: The original alias name is restored to the form, and the form is redisplayed with this message without further processing.

User Response: You can modify the alias by appropriate DELETE and ADD actions.

DBD3903 E RELATED ENTITY IDENTIFIER MAY NOT BE MODIFIED

Explanation: The user has attempted to modify an existing related entity identifier. The interactive display forms facility does not support this modification.

System Action: The original related entity subject name is restored to the form, and the form is redisplayed without further processing.

User Response: You can modify a related entity identifier via appropriate DELETE and ADD actions.

DBD3904 I MORE ALIASES EXIST

Explanation: There are more aliases than can fit in the "window." They cannot be displayed until some of the aliases displayed are deleted.

System Action: Processing continues.

User Response: Delete some of the aliases displayed, or use the command language if you want to see the other

aliases.

DELETE STRUCTURE (DBD41NN T)

DBD4101 I END OF DELETE_STRUCTURE

Explanation: DELETE_STRUCTURE execution has ended.

System Action: Execution continues with the next command.

User Response: Examine the messages issued by DELETE_STRUCTURE processing to verify that the entire structure was deleted. If preceding messages indicate a premature end of processing, note the subjects successfully deleted and reenter the command referencing the remaining structure subset.

DBD4102 I subject category (subject name) AND ALL ALIASES HAVE BEEN DELETED

Explanation: The subject named in the message has been deleted from the structure.

User Response: Verify that this subject does belong to the structure intended for deletion.

System Action: DELETE_STRUCTURE execution continues.

DBD4103 S ERROR REPOSITIONING IN TABLE-SUBJECT CATEGORY subject category subject code

Explanation: The subject category specified in the message was found in the hierarchy table previously, but cannot be located now. There is probably a program error.

System Action: Execution ends for the current command. Subsequent commands are only checked for syntax errors unless FLUSH=NO.

User Response: Note the subjects successfully deleted, and reenter the command referencing the remaining structure subset. If the problem persists, notify your IBM representative.

DBD4104 E INVALID USE OF STOP OPERAND

Explanation: The STOP operand must specify a category that is on the structure's delete path. This message will also be issued when STOP references a category that is not supported by the DELETE_STRUCTURE command, such as DDUSER, ATTRTYPE, or some installation-defined category.

System Action: Execution continues with the next command.

User Response: Refer to the discussion of the DELETE_STRUCTURE command in this manual and the DB/DC Data Dictionary Applications Guide.

DBD4105 W THE NON-STRUCTURAL (relkw) seqvalue RELATIONSHIP BETWEEN subject category (subject name) AND relcat (rsname) WAS DELETED

Explanation: The first subject named in the message participates in a relationship considered by DELETE_STRUCTURE to be NON-STRUCTURAL, with the second subject named in the message. These types of relationships are not considered as inhibitors to the delete structure process and are reported on for informational purposes only. See DBD0911 for more information on nonstructural relationships.

System Action: The relationship is deleted.

User Response: None required.

DBD4106 E INVALID SUBJECT CATEGORY FOR DELETE STRUCTURE

Explanation: The command has referenced a subject category such as DDUSER, ATTRTYPE, or some installation-defined category. These categories are not supported by DELETE_STRUCTURE.

System Action: The command is rejected.

User Response: Resubmit the command with proper category specified, or use the DELETE command to perform subject deletion.

RELOCATE (DBD42NN T)

DBD4201 E NO RELATIONSHIP EXISTS BETWEEN subject a/subject c AND SUBJB

Explanation: The relationship mentioned is necessary for the execution of this command, but does not exist.

System Action: Execution is terminated for this command. Subsequent commands are only checked for syntax errors unless FLUSH=NO.

User Response: Check for spelling errors. If necessary, add the appropriate relationship using the Update commands.

DBD4202 E SUBJC IS A DEPENDENT OF SUBJA

Explanation: The relocation specified cannot be made because SUBJECT C (the segment before which the data is to be moved) is part of the data to be moved (SUBJECT A and dependents).

System Action: The requested action is not performed. Subsequent commands are only checked for syntax errors unless FLUSH=NO.

User Response: Reenter the command correctly.

DBD4203 E SEGMENT(S) CURRENTLY IN SPECIFIED LOCATION

Explanation: The segments are already in the location to which they were to be moved.

System Action: Relocation is not done. Subsequent commands are only checked for syntax errors unless FLUSH=NO.

User Response: If another action was intended, reenter the command in correct format.

DBD4204 E REGION SIZE INSUFFICIENT FOR NUMBER OF DEPENDENTS

Explanation: The storage available to the program is too small to contain all dependents.

System Action: Relocation is not done. Subsequent commands are only checked for syntax errors unless FLUSH=NO.

User Response: Resubmit the job specifying a larger region size, or use several RELOCATE commands instead of one to move the segments.

DBD4205 E SUBJA IS THE SAME AS SUBJC

Explanation: The segment to be moved is the same as the segment before which it is to be relocated.

System Action: Relocation is not done. Subsequent commands are only checked for syntax errors unless FLUSH=NO.

User Response: Correct the command, and reenter it.

DBD4206 T UNEXPECTED STORAGE MANAGER RETURN CODE - nn

Explanation: The Storage Manager returned an unexpected value in register 15. There is probably a program error.

System Action: Execution is terminated.

User Response: Notify your IBM representative.

DBD4207 I SEGMENT subject name BEING RELOCATED

Explanation: The processing required to relocate the named segment has begun.

System Action: Execution continues.

User Response: No response is required.

DBD4208 I SEGMENT subject name HAS BEEN RELOCATED

Explanation: The relocation of the named segment has been completed.

User Response: No response is required.

System Action: Execution continues.

DBD4209 I RELOCATE PROCESSING COMPLETE

Explanation: All processing for the RELOCATE command has been completed.

User Response: No response is required.

System Action: Execution continues with the next command.

DBD4210 E DUMMY SEGMENT EXISTS - PRIOR RELOCATE INCOMPLETE

Explanation: The last RELOCATE command did not run to completion.

System Action: Relocation is not done. Subsequent commands are only checked for syntax errors unless FLUSH=NO.

User Response: Refer to the description of the RELOCATE command for recovery instructions.

DBD4211 W SEGMENT name - CALCULATED LEVEL VALUE IS ABOVE 15 - LEVEL SET TO 0. DBS HIERARCHY HAS TOO MANY LEVELS.

Explanation: During calculation of segment level values, the segment specified was found to have an implied level value greater than the maximum number of levels allowed in DL/I, which is 15.

System Action: The level number of this segment is set to 0 and execution continued.

User Response: Either the level value specified in the RELOCATE command was higher than desired, or the data base structure resulting from the specified relocation has resulted in an invalid data base structure. In either case, further RELOCATE commands can be used to produce the desired structure.

GIS SUPPORT (DBD43NN T)

DBD4301 E name DATABASE NOT FOUND IN DICTIONARY

Explanation: A command requested a DDT output for this database, but it could not be found in the Dictionary.

System Action: Execution ends for this command, but any subsequent commands are processed.

User Response: Verify that the command was correctly specified, especially the status (production/test) and DBD type (physical/logical) codes. If necessary, use DBD_IN to add the data base to the Dictionary.

DBD4302 W name PCB NOT FOUND IN DICTIONARY

Explanation: The command specified a PCB name, but it could not be found in the Dictionary.

System Action: Execution of the command continues as if no PCB name had been specified.

User Response: Verify that the PCB name was correctly specified. If necessary, use PSB_IN to add the PCB to the Dictionary.

DBD4303 I END OF DDT_OUT PROCESSING

Explanation: Processing of the current DDT_OUT command has completed.

System Action: Execution of the input command stream continues.

User Response: None.

DBD4304 E name DATABASE HAS NO SEGMENTS

Explanation: The data base specified has no relationships to any segments in the Dictionary.

System Action: Execution is terminated for the current command, any subsequent command is processed.

User Response: Verify that the command was correctly specified, especially the status (production/test) and DBD type (physical/logical) codes. If necessary, use DBD_IN to add the data base to the Dictionary.

DBD4305 W name DATABASE HAS NO DATASET JCL

Explanation: The data base specified has no User Data segments containing the mandatory JCL describing the data base data set(s).

System Action: Execution continues.

User Response: Add the required JCL to the data base User Data segment (DBSUSERn) and/or specify the number (1-5) of the segment and the number (0-999) of the first occurrence.

DBD4306 E MULTIPLE SPECIFICATION OF EDIT TYPE

Explanation: The user has requested two or more edit types for a single edit mode on the Reports Field screen.

System Action: The request is rejected and the screen regenerated.

User Response: Correct the edit type request to indicate only one type, and reenter the screen.

DBD4307 E RNGE OR PICT USED ONLY WITH VALIDATE

Explanation: The user has requested an edit type of RNGE or PICT for an edit mode other than Validate (Decode or Encode).

System Action: The request is rejected and the screen regenerated.

User Response: Correct the request to indicate a valid pair of edit type and edit mode values, and reenter the screen.

DBD4308 E EXIT ROUTINE NAME REQUIRED

Explanation: The user has specified the edit type of EDIT but has not supplied the user exit routine name.

System Action: The request is rejected and the screen regenerated.

User Response: Change the edit type or specify the exit routine name, and reenter the screen.

BATCH INPUT FORMS (DBD44NN T)

DBD4400 E INVALID OR NO USER-DATA NUMBER ON HEADER RECORD

Explanation: Either an invalid or no user data set identification number was provided on the header card for the input form.

System Action: The processing of this input form is terminated.

User Response: Enter a valid user data set identification number into the header card on the input form, and resubmit the form for processing.

DBD4401 E INVALID OR NO SUBJECT TYPE ON HEADER RECORD

Explanation: Either an invalid or no subject type code was provided in the header record for the input form.

System Action: The processing of this input form is terminated.

User Response: Enter a valid subject type code into the header card on the input form, and resubmit the form for processing.

DBD4402 E SUBJECT NAME MISSING ON HEADER RECORD

Explanation: No subject name was provided on the header card of the input form.

System Action: The processing of the input form is terminated.

User Response: Enter the correct Dictionary subject name in the header card on the input form, and resubmit the form for processing.

DBD4403 E CODE IN SUBJECT NAME NOT VALID FOR SPECIFIED SUBJECT TYPE

Explanation: The subject code that appears in the Dictionary subject name specified on the header card of the input form is inconsistent with the specified subject type.

System Action: The processing of the input form is terminated.

User Response: Correct the code value on the header card of the input form, and resubmit the form for processing.

DBD4404 E UNKNOWN FORM TYPE

Explanation: The form type (positions 5-7 of the input record) is not PLI, USR, DSC, SEG, or FLD.

System Action: The processing of this form is terminated. The remaining records of the form are read and ignored.

User Response: Correct the input, and resubmit the form for processing.

DBD4405 E MISSING OR INVALID NEW/OLD CODE ON HEADER RECORD

Explanation: Either an invalid or no OLD/NEW code was specified in the header card of the input form.

System Action: The processing of the input form is terminated.

User Response: Enter the correct OLD/NEW code in the header card of the input form, and resubmit the form for processing.

DBD4406 E "OLD" SPECIFIED, BUT DEFINITION DOES NOT EXIST

Explanation: The subject with the subject name specified in the header card exists in the Dictionary.

System Action: The processing of the input form is terminated.

User Response: Correct the subject name, subject type code, or the OLD/NEW code, and resubmit the form for processing.

DBD4407 E "NEW" SPECIFIED, BUT DEFINITION ALREADY EXISTS

Explanation: A definition already exists for a subject having the subject name specified on the header card of the input form.

System Action: The processing of the input form is terminated.

User Response: Correct the subject name, subject type, or OLD/NEW specification on the header card for the input form, and resubmit the form for processing.

DBD4408 E FIELD NOT NEW, TYPE AND BYTES NOT ADDED

Explanation: TYPE and/or BYTES was specified for a field that already exists in the Dictionary.

System Action: No command is issued to add or change the TYPE or BYTES attributes.

User Response: Update the TYPE and/or BYTES attributes with the command language.

DBD4410 E LINE NUMBER MISSING OR INVALID

Explanation: The line number on the description card being processed is either blank or not in the range of 1 through 255.

System Action: No further processing is done for the card. Processing continues with the next input record.

User Response: Correct the line number, and resubmit the card on a new batch input form.

DBD4411 E PLI TEXT-TYPE MISSING OR INVALID

Explanation: The PL/I text-type code, on the PL/I data card currently being processed is either blank or not C, N, or X.

System Action: No further processing is done for the card. Processing of the input form continues with the next input record.

User Response: Correct the PL/I text-type code, and resubmit the card on a new batch input form.

DBD4412 E INVALID DEFAULT STATUS CODE

Explanation: The status default specified on the batch form header record is invalid. The valid values are P, T, and 0 through 9.

System Action: The subject specified in the header record is not added, even if it is new. The remainder of the batch input form is flushed.

User Response: Correct the value, and resubmit the batch input form.

DBD4413 E INVALID DEFAULT LANG (CODE) VALUE

Explanation: The language code default specified on the batch form header card is invalid. The valid values are A, B, C, J, K, and L.

System Action: The subject specified in the header card is not added, even if it is new. The remainder of the batch form is flushed.

User Response: Correct the value, and resubmit the batch input form.

DBD4414 E INVALID DEFAULT DATABASE-TYPE CODE

Explanation: The data base type code default specified on the batch form header card is invalid. The valid values are L, P, X, Y, and F.

System Action: The subject specified in the header card is not added, even if it is new. The remainder of the batch form is flushed.

User Response: Correct the value, and resubmit the batch input form.

DBD4415 E INVALID DEFAULT OCCURRENCE NUMBER

Explanation: The occurrence number default specified in the batch form header card contains a nonnumeric character, an embedded blank, or is greater than 255.

System Action: The subject specified in the header card is not added even if it is new. The remainder of the batch form is flushed.

User Response: Correct the value, and resubmit the batch input form.

DBD4416 E DATA PRESENT IN UNUSED FIELD

Explanation: Data has been detected in a field that should be blank on the input card. Possibilities include the

"Type" field when processing User Data or Description text on the TEXT form and the "User Data No." field when processing a TEXT form header card requesting PL/I data or description

System Action: Processing of the input card is terminated.

User Response: Correct the input card, and resubmit the batch form.

DBD4417 E PLI TEXT DATA VALID ONLY FOR FIELD OR SEGMENT SUBJECTS

Explanation: An attempt has been made to enter PL/I data for subject that is not in the FIELD or SEGMENT category.

System Action: No further processing is done for the card. Processing of the input form continues with the next input record.

User Response: Reenter the PL/I data card on a new batch form that specifies a category of either FIELD or SEGMENT.

DBD4418 E NUMERIC FIELD CONTAINS NON-NUMERIC CHARACTER OR IMBEDDED BLANK

Explanation: A numeric field contains a nonnumeric character or an embedded blank.

System Action: The batch form card is not processed. The remainder of the batch form is flushed with a syntax check only unless FLUSH=NO.

User Response: Correct the field, and resubmit the cards that have not been processed.

DBD4420 E MISSING SUBJECT NAME ON "AKA" RECORD

Explanation: No alias subject name was specified on the preceding AKA input record.

System Action: Processing of the input form continues with the next input record.

User Response: Enter the missing alias, with a subject name, on a new batch input form.

DBD4425 E STATUS CODE status-code NOT AUTHORIZED FOR UPDATE

Explanation: The definition of a subject with the specified status-code is not authorized.

System Action: Processing is terminated.

User Response: Select an authorized status code, and resubmit the cards that have not been processed.

DBD4426 E CATEGORY SELECTED NOT AVAILABLE FOR USE

Explanation: Subjects in the selected category may not be updated by the current user.

System Action: Processing is terminated.

User Response: Ask the security administrator at your facility to place the category name into your security profile. Then resubmit the cards that have not been processed.

DBD4430 E MULTIPLE ATTRIBUTE RECORDS DETECTED

Explanation: More than one BSA or BPA record was included in the input form.

System Action: The data values contained in the second ATR record are ignored. Processing of the input form continues with the next input record.

User Response: If you wish to change one or more of the attribute values associated with the subject definition, you should do so using either the Dictionary command language or the Display-Forms facility.

DBD4431 E MULTIPLE "CBL" RECORDS DETECTED

Explanation: More than one CBL attribute record has been included in the input form.

System Action: The COBOL attribute values specified on the second CBL record are ignored. Processing of the input form continues with the next input record.

User Response: If you wish to change one or more of the COBOL attribute values associated with the subject definition, you should do so using either the Dictionary command language or the Display-Forms facility.

DBD4432 E MULTIPLE "OCCURS" SPECIFIED

Explanation: An OCCURS value has been specified on multiple OCC records or in an OCC record and in a CBL record.

System Action: The OCCURS value specified in the second record is ignored. If this record is a CBL record, the entire record is not processed but it is syntax checked. Processing of the input form continues with the next input record.

User Response: If you wish to change the OCCURS value, you should do so using either the Dictionary command language or the Display-Forms facility.

DBD4433 E MULTIPLE "VALUE" SPECIFIED

Explanation: A VALUE value has been specified on multiple VAL records or in a VAL record and in a CBL record.

System Action: The VALUE value specified in the second record is ignored. If this record is a CBL record, the entire record is not processed but is syntax checked. Processing of the input form continues with the next input record.

User Response: If you wish to change the VALUE value, you should do so using either the Dictionary command language or the Display-Forms facility.

DBD4434 E MULTIPLE PL/I ATTRIBUTE RECORDS SPECIFIED

Explanation: More than one PAT record was included in the input form.

System Action: The data values contained in the second PAT record are ignored. Processing of the input form continues with the next record.

User Response: If you wish to change one or more of the attribute values associated with the subject definition, you should do so using either the Dictionary command language or the Display-Forms facility.

DBD4435 E MULTIPLE PL/I PICTURE AND INITIAL RECORDS SPECIFIED

Explanation: More than one PPI record was included in the input form.

System Action: The data values contained in the second PPI record are ignored. Processing of the input form continues with the next record.

User Response: If you wish to change the picture or initial value associated with the subject definition, you should do so using either the Dictionary command language or the Display-Forms facility.

DBD4440 E MISSING OR INVALID NEW/OLD CODE ON "SFD" RECORD

Explanation: Either an invalid or no NEW/OLD code was specified on the preceding subfield record.

System Action: The data specified in the subfield record is ignored. Processing of the input form continues with the next input record.

User Response: Correct the indicated SFD record, and resubmit it on a new input form.

DBD4441 E MISSING SUBJECT NAME ON "SFD" RECORD

Explanation: No subject name for the subfield was specified in the preceding subfield record.

System Action: The data contained in the indicated subfield record is ignored. Processing of the input form continues with the next input record.

User Response: Correct the indicated SFD record, and resubmit it on a new input form.

**DBD4442 E MISSING OR INVALID
SUBSTART VALUE ON "SFD"
RECORD**

Explanation: Substart value must be 1 through 32767.

System Action: Processing of the SFD record is terminated. If FLUSH=YES, the remainder of the batch form is syntax checked only.

User Response: Correct the error, and resubmit the records that have not been processed.

**DBD4443 E BIT SUBSTART MAY NOT BE
SPECIFIED FOR COBOL
SUBFIELDS**

Explanation: A bit substart position was specified for a COBOL subfield.

System Action: The data specified in the subfield record is ignored. Processing of the input form continues with the next record.

User Response: Correct the indicated SFD record, and resubmit it on a new input form.

**DBD4450 E MISSING OR INVALID NEW/OLD
CODE ON "RFD" RECORD**

Explanation: Either an invalid or no NEW/OLD code was specified in the preceding related-field record.

System Action: The data specified in the indicated record is ignored. Processing of the input form continues with the next input record.

User Response: Correct the RFD record, and resubmit it on a new input form.

**DBD4451 E MISSING RELATED FIELD
SUBJECT NAME ON "RFD"
RECORD**

Explanation: No subject name for the related field was specified in the preceding related field record.

System Action: The data specified in the indicated record is ignored. Processing of the input continues with the next input record.

User Response: Correct the indicated RFD record, and resubmit it on a new input form.

**DBD4452 E MISSING OR INVALID START
VALUE ON "RFD" RECORD**

Explanation: The start value must be 1 through 32767.

System Action: Processing of the RFD record is terminated.

User Response: Correct the error, and resubmit those records not processed.

**DBD4453 E BIT START MAY NOT BE
SPECIFIED FOR COBOL
RELATED FIELDS**

Explanation: A bit start position was specified for a COBOL related field.

System Action: The data specified in the related field record is ignored. Processing of the input form continues with the next record.

User Response: Correct the indicated RFD record, and resubmit it on a new input form.

**DBD4460 E EXPECTED CONTINUATION NOT
RECEIVED**

Explanation: The previous input record indicated that a continued OCCURS (OCC) or VALUE (VAL) record was to be supplied, but it has not been found.

System Action: This record and the previous record that indicates a continuation are ignored.

User Response: Correct these two records, and resubmit them on a new input form.

DBD4461 E UNRECOGNIZED RECORD TYPE

Explanation: The first three characters of the input record do not contain a valid code.

System Action: The input record is displayed, but it is ignored for processing purposes.

User Response: Correct the input record, and resubmit it on a new input form.

**DBD4463 E CODE MUST BE BLANK FOR AN
EXTENSIBILITY SUBJECT**

Explanation: A subject code was specified for a subject in an Extensibility category.

System Action: The subject specified in the header card is not added, even if it is new. The remainder of the batch form is flushed.

User Response: Correct the value, and resubmit the batch input form.

DBD4464 E CATEGORY NOT INSTALLED

Explanation: The Extensibility category specified on the header card has not been installed.

System Action: The subject specified in the header card is not added, even if it is new. The remainder of the batch form is flushed.

User Response: Install the category, and resubmit the batch input form.

DBD4470 I BATCH FORM PROCESSING SUCCESSFULLY COMPLETED

Explanation: The batch form processing has completed successfully.

System Action: Control is returned to the language processing portion of the Dictionary, and scanning of the input stream continues.

User Response: None.

DBD4471 W "DBDFEND" RECORD MISSING

Explanation: An end-of-file condition or a new header record was detected when more data or an end card was expected.

System Action: Processing of the current form is terminated. If an end-of-file condition was detected, control is returned to the language processing portion of the Dictionary to terminate the session. If a new header record was detected, the new form is processed.

User Response: If a report is required, submit a REPORT command.

DBD4472 I ERROR DETECTED -- BATCH FORM FLUSH IN PROGRESS

Explanation: An error was detected during the processing of a batch form. The error occurred during the syntax checking of a header card, during an attempt to update the Dictionary with header information, during the syntax checking of a batch form card with FLUSH=YES, or during an attempt to update the Dictionary with FLUSH=YES.

System Action: The remainder of the batch form is syntax checked only. A report is issued if requested on the DBDFEND card, unless the error occurred during header processing.

User Response: Correct the error, and resubmit the batch form cards that have not been processed.

DBD4473 I BATCH FORM PROCESSING COMPLETED WITH ERROR(S)

Explanation: At least one error was detected during the processing of the batch form. The error occurred during the syntax check of a batch form card or during an attempt to update the Dictionary.

System Action: Because FLUSH=NO, an attempt was made to process the remainder of the batch form as if no error had occurred.

User Response: Correct the error, and resubmit the cards that have not been processed.

DBD4474 I BATCH FORM FUNCTION NOT PERFORMED, SYNTAX CHECK ONLY

Explanation: This function was not completely performed as specified, but it may have been partially performed.

System Action: If there was an error in the previous batch form or command, the current batch form is only checked for the proper syntax.

User Response: Correct the batch form in error, and resubmit the cards that have not been processed.

DBD4480 E SCALING FACTOR MAY NOT BE SPECIFIED WITHOUT NUMBER OF DIGITS

Explanation: The scaling factor was specified without the number of digits on the PL/I attributes record.

System Action: The data specified in the PAT record is ignored. Processing of the input form continues with the next record.

User Response: Correct the PAT record, and resubmit it on a new input form.

DBD4481 E LOWER BOUND MAY ONLY BE SPECIFIED WITH UPPER BOUND

Explanation: A lower bound was specified for one of the three dimensions without the corresponding upper bound on the PL/I attributes record.

System Action: The data specified in the PAT record is ignored. Processing of the input form continues with the next record.

User Response: Correct the PAT record, and resubmit it on a new input form.

DBD4482 E INVALID PL/I FORMAT CODE SPECIFIED

Explanation: An invalid PL/I format code was specified on the PL/I attributes record. Valid format codes are BIN, DEC, BIT, PTR, and CHAR.

System Action: The data specified in the PAT record is ignored. Processing of the input form continues with the next record.

User Response: Correct the PAT record, and resubmit it on a new input form.

DBD4483 E INVALID PL/I OPTION CODE SPECIFIED

Explanation: An invalid PL/I option code was specified on the PL/I attributes record. Valid option codes are FIXED, FLOAT, and VAR.

System Action: The data specified in the PAT record is ignored. Processing of the input form continues with the next record.

User Response: Correct the PAT record, and resubmit it on a new input form.

DBD4484 E INVALID PL/I MODE CODE SPECIFIED

Explanation: An invalid PL/I mode code was specified on the PL/I attributes record. Valid mode codes are REAL and CPLX.

System Action: The data specified in the PAT record is ignored. Processing of the input form continues with the next record.

User Response: Correct the PAT record, and resubmit it on a new input form.

DBD4485 E INVALID PL/I ALIGNMENT CODE SPECIFIED

Explanation: An invalid PL/I alignment code was specified on the PL/I attributes record. Valid alignment codes are Y (YES) and N (NO).

System Action: The data specified in the PAT record is ignored. Processing of the input form continues with the next record.

User Response: Correct the PAT record, and resubmit it on a new input form.

DBD4486 E INITIAL CLAUSE NOT ENCLOSED IN PARENTHESES

Explanation: The initial clause entered on a batch forms Picture and Initial card (PPI) is not enclosed in parentheses.

System Action: Neither the picture nor the initial clause is added to the Dictionary. Processing of the input form continues with the next record.

User Response: Make certain that the initial clause begins in the proper column on the PPI card and that it is enclosed in parentheses. Resubmit the PPI record on a new form.

DBD4487 E PICTURE CLAUSE NOT ENCLOSED IN QUOTES

Explanation: The PICTURE clause entered on a batch forms Picture and Initial card (PPI) is not enclosed in quotation marks.

System Action: Neither the picture nor the initial clause is added to the Dictionary. Processing of the input form continues with the next record.

User Response: Make certain that the PICTURE clause begins in the proper column on the PPI card and that it is enclosed in quotation marks. Resubmit the PPI record on a new form.

DBD4490 E ERROR SEARCHING FOR SUBJECT

Explanation: A DL/I error was encountered while searching the Dictionary data bases. This error should not occur.

System Action: Processing of the input form is terminated.

User Response: Contact your local IBM representative.

EXTEND RELATIONSHIP (DBD45NN T)

DBD4500 I SOURCE type FOR EXTEND-RELATIONSHIP IS subject

Explanation: Identifies the source subject for an XR command being processed.

System Action: Execution of the command continues.

User Response: None.

DBD4501 I TARGET type FOR EXTEND-RELATIONSHIP IS subject

Explanation: This message identifies the target subject for the XR command being processed. The type is a one-character abbreviation for the target subject category and may be E (element), D (data base), P (PCB), S (segment), or Y (system object).

System Action: Execution of the command continues.

User Response: None.

DBD4502 I RELATIONSHIP TO category name CREATED

Explanation: This message identifies a field or segment that has been related to the target subject. It is issued for each field that is related.

System Action: Execution of the command continues.

User Response: None.

**DBD4503 I RELATIONSHIP TO FIELD name
CREATED, START= n**

Explanation: This message identifies a field that has been related to the target subject, where the target subject type is segment. It is issued for each field that is related to a target segment.

System Action: Execution of the command continues.

User Response: None.

**DBD4504 I RELATIONSHIP TO category
name ALREADY EXISTS**

Explanation: A relationship already exists between the source subject and the target subject specified in the XR command.

System Action: No new relationship is created. Command execution continues.

User Response: None.

**DBD4505 I EXTEND-RELATIONSHIP
FUNCTION COMPLETE**

Explanation: The XR command has completed execution successfully. This message marks the end of the output messages generated by the XR command processor.

System Action: Control is returned to process the next user input.

User Response: None.

**DBD4506 I TARGET DOES NOT EXIST,
WILL BE ADDED**

Explanation: The target subject for the EXTEND_RELATIONSHIP command does not exist in the Dictionary. With the first successful ADD_RELATIONSHIP generated by the EXTEND_RELATIONSHIP command processor, the target subject will be added to the Dictionary.

System Action: Execution continues.

User Response: None.

**DBD4510 E SOURCE subject FOR
EXTEND-RELATIONSHIP NOT
DEFINED**

Explanation: There is no definition in the Dictionary for the source segment or operand specified in the XR command.

System Action: Execution of the command is terminated.

User Response: Reenter the command, specifying the correct Dictionary subject name for the source segment or field.

**DBD4512 I EXTEND-RELATIONSHIP
FUNCTION NOT EXECUTED**

Explanation: The XR command was not executed because of the error condition indicated in a preceding message (no new relationships were created in the Dictionary).

System Action: Execution of the XR command is terminated. Control is returned to process the next user input.

User Response: Correct the error indicated by the preceding message, and reenter the command.

**DBD4513 I EXTEND-RELATIONSHIP
FUNCTION TERMINATED**

Explanation: XR command execution was terminated during execution because of the error condition indicated in a preceding message. Some new relationships, indicated by message DBD4502 or DBD4503, may have been created before execution was terminated.

User Response: Correct the error indicated by the preceding message, and reenter the command.

**DBD4514 E REQUIRED START VALUE
MISSING**

Explanation: The XR command that was issued requires a START= value and none was supplied.

System Action: Command execution is terminated with message DBD4512.

User Response: Reenter the XR command with a START= value.

**DBD4515 E SYSTEM ERROR error IN
MODULE module**

Explanation: A system error was encountered in an attempt to retrieve data from the Dictionary.

System Action: The XR execution is terminated.

User Response: Notify your IBM representative.

DBD4516 E INVALID SUBJECT CATEGORY

Explanation: The source for the XR command is not a segment or a field, or the target of the command is a field or an installation-defined category.

System Action: XR command execution is terminated.

User Response: Correct the source or target subject category, and resubmit the command.

DBD4517 E TARGET SUBJECT MUST BE SPECIFIED

Explanation: The target for the XR command has been omitted.

System Action: XR command execution is terminated.

User Response: Specify the target type (category) and subject name, and resubmit the command.

DBD4519 S DL/I ERROR ON DATABASE database STATUS WAS status LAST SEG WAS segment

Explanation: A DL/I error return code was encountered while performing a retrieval of related fields.

System Action: XR command execution is terminated.

User Response: Take whatever steps are necessary to correct the condition responsible for the error return code.

DBD4520 E RETRIEVAL FAILED: CONCURRENT USER INTERFERENCE SUSPECTED

Explanation: A group of subfields was being retrieved but one of the already retrieved subfields could no longer be found in the Dictionary.

System Action: XR command execution is terminated.

User Response: Reenter the XR command, determine which subfield was deleted from the Dictionary, and, possibly, issue a DELETE_RELATIONSHIP command for the relationship that was already created.

DBD4870 S INVALID CALL TO DBDDGCG- [Invalid Command Code | Invalid Connector | Not a Key Entry | Not a Subfield Entry | Not a Field or Constant | Cmdarea Overflow]

Explanation: A Dictionary module called the DBDDGCG module with invalid input for generating a command, or input that generated too large a command.

System Action: Command generation is terminated.

User Response: Notify your IBM representative.

DBD4880 E RETURN CODE nn FROM MODULE name

Explanation: A Dictionary command was generated and submitted for execution by some Dictionary function. This message indicates that the function invoked by the generated command was unable to complete the request. This typically can happen on a RECALCULATE_SEGMENT request, which results in the generation of several Dictionary commands. If the segment to be recalculated is not found, this message is issued.

System Action: The function in process is terminated.

User Response: Correct the external Dictionary command, and resubmit it for execution. The previous set of messages will usually identify the source of the problem.

DBD4890 S DICTIONARY MESSAGE PROBLEM HAS OCCURRED

Explanation: The Dictionary message analysis module, DBDDMAG, encountered an execution error while attempting to put a message in the response area on behalf of a display form processor.

System Action: This message is placed in the response area.

User Response: Analyze the sequence of steps last attempted on a display form. Correct any invalid data entered, and try the display form action again. If this fails, notify your IBM representative.

DBD4891 E MESSAGE TOO LONG: PLEASE REFER TO MESSAGE DBDXXXX

Explanation: An attempt was made to print message DBDxxxx, but it was found to be greater than 79 characters long.

System Action: Execution is terminated.

User Response: Refer to message DBDxxxx in this appendix for the text of the message and the explanation.

PUNCH (DBD49NN T)

DBD4900 S DL/I ERROR ON DATABASE database STATUS WAS status LAST SEG WAS segment

Explanation: A DL/I error return code was encountered while performing a retrieval operation from a Dictionary data base.

System Action: PUNCH command execution is terminated.

User Response: Take whatever steps are necessary to correct the condition responsible for the error return code.

DBD4901 I PUNCH SUBJECT HAS NO USER-DATA (n) SEGMENTS

Explanation: The punch subject has no user-data (n) segments.

System Action: Command execution is terminated.

User Response: None.

DBD4902 W PUNCH SUBJECT NOT IN DICTIONARY

Explanation: The punch subject is not in the Dictionary.

System Action: Command execution is terminated.

User Response: Correct the subject name, and resubmit the command.

DBD4903 I USER-DATA SEGMENT WITH ID IDENTIFICATION CONTAINS ALL BLANKS

Explanation: The user-data segment with the noted identification contains all blanks.

System Action: A blank line is punched, and execution continues.

User Response: None.

DBD4904 I USER-DATA ID WAS NOT SPECIFIED-IT HAS BEEN SET TO 1

Explanation: The UDNO (or U1 through U5) parameter was not specified.

System Action: Execution continues with UDNO = 1 assumed.

User Response: If UDNO = 1 was not intended, specify the USER-DATA number and resubmit the command.

DBD4906 W PUNCH DEST SET TO 'P' FOR BATCH MODE

Explanation: If the DEST= keyword is not specified in the input command, the default value is 'P'=PUNCH when running in batch mode.

System Action: Execution continues.

User Response: If a different output destination is desired, it must be explicitly specified in the input command.

IMS/VIS SYSTEM DEFINITION SUPPORT
(DBD50NN T)

DBD5000 E DL/I ERROR ON DATABASE database STATUS WAS status LAST SEG WAS segment

Explanation: A DL/I error return code was encountered while performing an operation against one of the Dictionary data bases.

System Action: STAGE_1_OUT command execution is terminated.

User Response: Take whatever steps are necessary to correct the condition responsible for the error return code.

DBD5001 E SUBJECT SYSDEF DOES NOT EXIST

Explanation: The subject system definition does not exist.

System Action: STAGE_1_OUT command execution is terminated.

User Response: Specify an existing SYSDEF subject name, and resubmit the command.

DBD5005 W SYSDEF name HAS NO RELATED DATABASES

Explanation: The named system definition subject is not related to any data base subjects.

System Action: A return code of 4 is set. Execution continues with the next command.

User Response: Add relationships between this system definition subject and one or more data base subjects.

DBD5006 W SYSDEF name HAS NO RELATED PSBS

Explanation: The named system definition subject is not related to any PSB subjects.

System Action: A return code of 4 is set. Execution continues with the next command.

User Response: Add a relationship between this system definition subject and one or more PSB subjects.

DBD5007 W SYSDEF name RELATED TO PSB name HAS NO RELATED INPUT TRANSACTIONS

Explanation: The named system definition subject which is related to the named PSB subject has no commonly related transaction. The SDF-TRN or the PSB-TRN (with OSTTYPE=I) relationship (or both) has not been established.

System Action: A return code of 4 is set. Execution continues with the next SDF-PSB relationship encountered or with the next input command.

User Response: Add the missing relationship(s), and rerun the job.

DEFAULT (DBD51NN T)

DBD5100 I default-variable value

Explanation: Execution of the DEFAULT command produces one of these messages for each default variable.

System Action: Execution of the input command stream continues.

User Response: None.

DBD5101 E LOG OPTION 'NO' IS REJECTED, SYSTEM OPTION IS 'YES'

Explanation: The commands SETLOGU and SETLOGI may not suppress logging when the system defaults specify that the corresponding class of commands (user or internal) are to be logged.

System Action: Execution of the input command stream continues.

User Response: None

CHECKPOINT (DBD52NN T)

DBD5200 I DL/I CHECKPOINT DDhhmmss COMPLETED

Explanation: An IMS/VS checkpoint has been successfully executed. The checkpoint identification is "DDhhmmss", where "hh" is hour, "mm" is minute, and "ss" is second.

System Action: Execution of the input command stream continues.

User Response: None. All inputs and modifications occasioned by the commands processed prior to this message have been incorporated in the Dictionary.

DBD5201 W IMS/VS IS CLOSING DOWN, THE DICTIONARY IS TERMINATING

Explanation: IMS/VS is in the process of closing down; any subsequent calls to DL/I would cause the Dictionary to be ABENDED.

System Action: The input command stream is flushed, and the Dictionary run is terminated without further execution.

User Response: All inputs and modifications occasioned by the commands processed prior to this message have been incorporated in the Dictionary. When IMS/VS is brought up again, the input commands which are flushed at this time should be resubmitted.

DBD5202 E ERROR OCCURRED DURING DL/I CHECKPOINT DDhhmmss, RETURN CODE xx, PROCESSING IS TERMINATED

Explanation: A failure has occurred during execution of an IMS/VS checkpoint. The return code from the checkpoint module was "xx".

System Action: The input command stream is flushed, and the Dictionary run is terminated without further execution.

User Response: The installation's IBM representative should be notified, and requested to analyze the reason for the checkpoint failure. All inputs and/or updates occasioned by the commands preceding this message (and following the last occurrence of message DBD5200) have been lost.

DBD5203 E ERROR OCCURRED DURING DL/I CHECKPOINT ckptid, RETURN CODE xx

Explanation: An error has occurred while attempting a DL/I checkpoint call in a CICS environment. xx is the status code returned by DL/I.

System Action: This message is logged by the Dictionary and also sent to the system console. Execution continues, but this message will probably continue to be issued after each attempted checkpoint call.

User Response: Notify the Data Base Administrator or other person responsible for the Data Dictionary, since checkpointing of vital database status is not occurring. To shut down the Dictionary environment, the DBA may issue a DDSD shutdown command. If checkpointing is not required, further occurrences of the message may be eliminated by issuing a SETCKPT=0 command on the Dictionary Command Screen. Program Access (DBD53nn t)

PROGRAM ACCESS (DBD53NN +)

DBD5305 S INVALID CALL TO PROGRAM ACCESS FACILITY FROM USER PROGRAM userpgm

Explanation: A user program called the Program Access facility but did not pass the address of an area

recognizable as a PACA.

System Action: Control is returned to the user load module with return code set in register 15.

User Response: Probable error in the user's program access routine.

DBD5306 S INVALID REQUEST CODE code PASSED TO DBDWGMAT

Explanation: Module DBDWGMAT was called with a nonretrieve request code. This is a "cannot happen" condition.

System Action: This message is followed by ABEND 3800.

User Response: Probable error in Dictionary routines. Contact your IBM representative.

CICS SUPPORT (DBD54NN T)

DBD5400 I DB/DC DATA DICTIONARY INITIALIZATION PROCESSING COMPLETED SUCCESSFULLY

Explanation: The Dictionary-supplied system initialization overlay module has completed its tasks successfully. When CICS system initialization is completed, Dictionary work may be initiated.

System Action: Execution continues in the normal CICS initialization process.

User Response: None. Informational message.

DBD5405 I DB/DC DATA DICTIONARY WILL NOT BE AVAILABLE, REPLY "GO" OR "CANCEL"

Explanation: A previous error message details the reason why the Dictionary will not be available. The console operator is given the opportunity to terminate the CICS initialization process or continue.

System Action: If a reply of "GO" is received, execution continues as if the initialization was successful. If a reply of "CANCEL" is received, the partition or region will be terminated.

User Response: Reply "GO" or "CANCEL" depending on whether or not Dictionary work will be required for this CICS session.

DBD5406 E TASK IS NOT APF-AUTHORIZED

Explanation: The operating system task is not operating in an "authorized" mode. Possible reasons for this

include DFHSIP was not linked into an "authorized" library with a SETCODE AC(1) specification (SVS, MVS, or VS1), or DBDHOSTA was not linked into an "authorized" library (SVS or VS1).

System Action: Message DBD5405 is issued.

User Response: If Dictionary work is required, make sure that all modules that must reside in "authorized" libraries are linked into the proper library correctly.

DBD5407 E (E)STAE MACRO FAILED, REASON CODE n

Explanation: The (E)STAE macro could not be issued successfully. Possible reasons include 10 - unexpected error, and 14 - storage unobtainable for SCB.

System Action: Message DBD5405 is issued.

User Response: Determine why the (E)STAE macro could not be issued, and correct the problem.

DBD5410 I DICTIONARY START-UP SUCCESSFULL

Explanation: The environment for communication with the Dictionary region or partition has been initialized "on the CICS side." Dictionary sessions may now be attempted.

System Action: None.

User Response: None.

DBD5411 I FORCED SHUTDOWN COMPLETE

Explanation: The Dictionary environment shut down requested through the MODE=FORCE parameter of the shut down transaction has been completed.

System Action: The Dictionary environment has been shut down. All active sessions will receive message DBD5429, unless CICS is terminated before the message can be sent.

User Response: Presumably, this transaction was entered in preparation for an "immediate" shutdown of CICS. The CICS shutdown may now be requested.

DBD5412 I type SHUTDOWN IN PROCESS, CURRENTLY n USERS

Explanation: The "normal" or "immediate" shutdown procedure is proceeding normally. If the type is NO, then shutdown has not yet been requested.

System Action: Processing continues.

User Response: None required. The message is intended to let the submitter know how many Dictionary sessions are in process.

**DBD5413 E UNEXPECTED RETURN CODE n
FROM FUNCTION xxx IN
MODULE yyy**

Explanation: If the n is 48, xxx is CONNECT, and yyy is DBDHDICT, then the problem is probably caused by an "immediate" shutdown of CICS followed by a startup of CICS before the batch Dictionary region has terminated and been restarted.

System Action: Execution of the appropriate Dictionary session or the entire Dictionary environment is terminated.

User Response: If the problem is as described above, then cancel the batch region, restart the batch region, and reissue the transaction to begin a Dictionary session. Otherwise, an internal Dictionary problem may be indicated. Notify your IBM representative.

DBD5414 I SHUTDOWN MODE CHANGED

Explanation: The Dictionary environment shutdown transaction has been entered with MODE=IMM. A prior transaction had initiated the shutdown process with MODE=NORMAL, but now the MODE will be changed to IMM.

System Action: The shutdown process continues, but now in an "IMM" mode rather than "NORMAL" mode. This means that terminal operators involved in Dictionary sessions will receive one more screen of data, but will not be permitted to enter anything else.

User Response: None required. This is purely an informational message.

**DBD5415 E TEMPORARY STORAGE FACILITY
NOT AVAILABLE**

Explanation: The CICS system being run contains the "dummy" Temporary Storage Program. Dictionary operation requires a nondummy version of this module.

System Action: Execution is terminated, and no Dictionary facilities will be available.

User Response: If Dictionary work is required, terminate CICS and bring it up again with a nondummy Temporary Storage Program.

**DBD5416 E UNABLE TO INITIALIZE
ENVIRONMENT, REASON CODE n**

Explanation: The LOGON (OS) or DEFINE (DOS) operation failed and further interregion communication is impossible. In an OS environment, if the reason code is 'A2', it normally indicates that DFHSIA2 has not been run. If the reason code is 'A3', it normally indicates that DFHSIA2 has been run, but a problem has been found and the console operator replied 'GO' to message DBD5405. In a DOS environment, a reason code of 8 means that the XECB table is full.

System Action: The startup transaction is terminated.

User Response: The LOGON (OS) or DEFINE (DOS) operation failed and further interrogation If DFHSIA2 has not been run check the SIMODS parameter of the SIT used in this startup of CICS.

**DBD5417 W ENVIRONMENT ALREADY
INITIALIZED**

Explanation: The startup transaction has already been invoked and processed successfully.

System Action: The transaction is terminated.

User Response: None required. Dictionary sessions may be initiated.

**DBD5418 I DB/DC DATA DICTIONARY
READY FOR COMMUNICATION
WITH CICS**

Explanation: This message is written to the operating system console when the Dictionary region or partition is available.

System Action: The Dictionary waits for communication from the CICS region/partition.

User Response: No response required.

**DBD5419 E type-1 SHUTDOWN REJECTED,
type-2 SHUTDOWN IN PROCESS**

Explanation: The shutdown transaction was invoked with MODE=type-1 when a MODE=type-2 shutdown was in process.

System Action: The transaction is terminated.

User Response: Variable based on circumstances. If the required shutdown is already in process, no response is required.

**DBD5421 W DICTIONARY ENVIRONMENT
SHUTDOWN IN PROCESS,
TRANSACTION TERMINATED**

Explanation: A shutdown of the Dictionary environment has been requested prior to the invocation of this task and therefore no new

Dictionary sessions may be started.

System Action: The transaction is terminated.

User Response: When the Dictionary environment may be brought-up again, do so and resubmit this transaction.

DBD5422 E INVALID MODE VALUE - nnnn

Explanation: The input specified for the MODE parameter on the shutdown transaction is invalid. Valid specifications are NORMAL, INQUIRE, IMM, and FORCE.

System Action: The transaction is terminated.

User Response: Reenter the shutdown transaction with a correct MODE specification.

DBD5425 E UNABLE TO LOAD CONTROL BLOCK/MAP DBDHyy, TRANSACTION TERMINATED

Explanation: The module containing the necessary control blocks and maps to process the current screen could not be loaded. The most likely reason for this is that it was not included in the PPT.

System Action: The transaction is terminated.

User Response: Fix the cause of the problem, and resubmit the transaction.

DBD5426 E ENVIRONMENT HAS NOT BEEN INITIALIZED, "START-UP" TRANSACTION IS REQUIRED

Explanation: The Dictionary startup transaction has not been initiated and the Dictionary environment has not been initialized.

System Action: The transaction is terminated.

User Response: Invoke the startup transaction, and then resubmit this transaction.

DBD5427 E INVALID MOD NAME SPECIFIED, TRANSACTION TERMINATED

Explanation: The MOD= parameter did not contain the name of a valid Header screen.

System Action: The transaction is terminated.

User Response: Specify a correct MOD value or let it default to HDRMOD.

DBD5428 E DICTIONARY NOT YET STARTED, TRANSACTION TERMINATED

Explanation: The Dictionary has not been started in the batch region or partition and therefore communication cannot be established.

System Action: The transaction is terminated.

User Response: Start the Dictionary in the batch region or notify whomever is responsible for starting it. After it has been started, resubmit this transaction.

DBD5429 I IMMEDIATE DICTIONARY SHUTDOWN IS REQUESTED, TRANSACTION TERMINATED

Explanation: A user at another terminal has requested an immediate shutdown of the Dictionary environment.

System Action: The Dictionary session is terminated, and no further session may be initiated until the Dictionary environment is started again (via the startup transaction).

User Response: None required. This is purely an informational message.

DBD5430 I NO STORAGE, OUTPUT SEGMENTED, PRESS ENTER TO CONTINUE

Explanation: The transaction was unable to GETMAIN enough space to contain the complete Dictionary response to the previous command.

System Action: The transaction waits for the user response.

User Response: Press ENTER to display the first segment of the response. Normal actions should be used to page through this first segment until message DBD5431 is displayed. To subsequently display the remainder of the response, press ENTER while any page of the segment is displayed. WARNING: No other user request will be serviced until the complete response is received from the Dictionary.

DBD5431 I END OF SEGMENT - ENTER YES TO REPEAT, NO TO DISPLAY NEXT SEGMENT

Explanation: A large command response has been received from the Dictionary and has been segmented (see DBD5430). The user has just displayed the last screen of the current segment and has requested the display of the next screen.

System Action: The transaction waits for a user response.

User Response: Enter YES to redisplay the current segment, starting with the first screen, or NO to purge this segment and display the next.

**DBD5435 I DICTIONARY HAS ABENDED,
TRANSACTION TERMINATED**

Explanation: The Dictionary, running in the batch region or partition, has terminated abnormally. No further communication with the Dictionary is possible.

System Action: The Dictionary session is terminated.

User Response: Correct the cause of the Dictionary ABEND, restart the Dictionary in the batch region, reenter the Dictionary startup transaction, and reinitiate any Dictionary sessions.

**DBD5440 I CICS ABEND DETECTED, DB/DC
DATA DICTIONARY NOTIFIED**

Explanation: A CICS region or partition ABEND has been intercepted and the Dictionary has been notified so that it may terminate itself. This message will appear on the OS/VS system operator's console when the online region's (E)STAE exit is entered.

System Action: The CICS region ABEND continues.

User Response: None.

**DBD5441 I DB/DC DATA DICTIONARY
ABEND DETECTED, CICS
NOTIFIED**

Explanation: A Dictionary region or partition ABEND has been intercepted and CICS has been notified so that it doesn't try to proceed with further communication. This message will appear on the OS/VS system operator's console when the batch region's (E)STAE exit is entered.

System Action: The Dictionary region ABEND continues.

User Response: None.

**DBD5445 E UNEXPECTED RETURN CODE n
FROM FUNCTION xxx IN
MODULE yyy (DB/DC DATA
DICTIONARY)**

Explanation: This message appears on the system operator's console when an error occurs.

System Action: Processing is terminated.

User Response: This message indicates some internal Dictionary problem. Notify your IBM representative.

**DBD5455 E (E)STAE MACRO FAILED,
REASON CODE n (DB/DC DATA
DICTIONARY)**

Explanation: An unexpected return code was returned following a STAE or ESTAE macro instruction.

System Action: The job is terminated, and the batch Dictionary is not started.

User Response: Correct the problem, and restart the job in order to be able to do any Dictionary work in CICS.

**DBD5456 E TASK IS NOT APF-AUTHORIZED
(DB/DC DATA DICTIONARY)**

Explanation: The operating system task is not operating in an "authorized" mode. Possible reasons for this include DBDHRCCI (or DBDHRCCC) was not linked into an "authorized" library with a SETCODE AC(1) specification (SVS, MVS, or VS1), or DBDHBSTA was not linked into an "authorized" library (SVS or VS1).

System Action: The job is terminated, and the batch Dictionary is not started.

User Response: If Dictionary work is required, make sure that all modules that must reside in "authorized" libraries are linked into the proper library correctly, and restart the batch region.

**DBD5460 I DB/DC DATA DICTIONARY
SHUTDOWN REQUESTED**

Explanation: The batch Dictionary region has been notified that a shutdown of the environment with CICS has been requested by a terminal operator in the CICS region.

System Action: The batch region is terminated.

User Response: None.

**DBD5466 E UNABLE TO INITIALIZE DB/DC
DATA DICTIONARY
ENVIRONMENT, REASON CODE n**

Explanation: The LOGON (OS) or DEFINE (DOS) operation failed and further interregion communication is impossible. In an OS environment, if the reason code is 'A2', it normally indicates that one of the Dictionary-supplied region controllers, DBDHRCCI or DBDHRCCC, has not been executed. In a DOS environment, a reason code of 8 means that the XECB table is full.

System Action: The startup transaction is terminated.

User Response: Correct the cause of the problem, and resubmit the startup transaction.

INSTALL (DBD55NN T)

DBD5500 I INSTALLATION OF type name COMPLETED

Explanation: This message is issued at the end of INSTALL command execution to indicate that installation of the indicated subject category or relationship-type has been completed successfully. The variable "type" will be either of the keywords CATEGORY or RELTYPE, according to whether the message was issued in response to an INSTALL CATEGORY or INSTALL RELTYPE command. The Dictionary subject name for the category or reltype definition will appear in place of the variable "name".

System Action: Installation execution for the category or relationship-type is completed.

User Response: None.

DBD5501 E type name NOT INSTALLED BECAUSE OF ERROR CONDITION

Explanation: This message is issued at the end of INSTALL command execution to indicate that the subject category or relationship-type definition was not installed because of one or more errors encountered during the installation process. The error(s) are indicated by some preceding message(s). The variable "type" will be either of the keywords CATEGORY or RELTYPE, as appropriate. The Dictionary subject name for the category or reltype definition will appear in the message in place of the variable "name".

System Action: None.

User Response: The error(s) indicated in the preceding message(s) must be corrected before the category or reltype definition may be installed.

DBD5502 I type name NOT INSTALLED (UPDATE=NO)

Explanation: This informational message is issued at the end of INSTALL command execution to indicate that the subject category or relationship-type definition was not actually installed because of the UPDATE=NO option specified on the INSTALL command. The variable "type" will be either of the keywords CATEGORY or RELTYPE. The Dictionary subject name for the category or reltype definition will appear in the message in place of the variable "name".

System Action: None.

User Response: If the user wishes to have the category or relationship-type installed in the Dictionary, the UPDATE=YES option must be specified on the INSTALL command.

DBD5503 I ATTRTYPE ALREADY INSTALLED

Explanation: This informational message is issued during the installation execution for a subject category or relationship-type to indicate that one of the attribute-types associated with the category or reltype definition already has "Installed" status.

System Action: None.

User Response: None.

DBD5504 I INSTALLATION SUCCESSFUL FOR ATTRTYPE name

Explanation: One instance of this informational message is issued at the end of the installation execution report (when UPDATE=YES has been specified on the INSTALL command) for each attribute-type definition that was "installed" as a part of the installation of the category or relationship-type. The variable "name" will be replaced in the message by the user-name in the subject name for the attribute-type definition.

System Action: The specified attribute-type definition is automatically changed to "Installed" status.

User Response: None.

DBD5505 E OBJECT OF INSTALL COMMAND NOT AN ECI SUBJECT

Explanation: Subject specified on the INSTALL command must be a CATEGORY or a RELTYPE.

System Action: Execution is terminated.

User Response: Correct the error, and reexecute INSTALL.

DBD5510 W STATUS OF type MUST BE 'P' FOR INSTALLATION

Explanation: The user has entered an INSTALL command with the UPDATE=YES option specifying a subject category or relationship-type definition that does not have "Production" status. The variable "type" will be either of the keywords CATEGORY or RELTYPE, as appropriate.

System Action: Installation execution is continued, but the subject category or relationship-type will not be installed (see message DBD5501).

User Response: The subject category or relationship-type definition must be copied to "Production" status before it may be installed.

**DBD5511 E REQUIRED ATTRIBUTE
 attribute NOT SPECIFIED**

Explanation: No value has been specified for a required attribute of a subject category, relationship-type, or attribute-type definition. The keyword identifying the attribute appears in place of the variable "attribute" in the message.

System Action: Installation execution is continued, but the subject category or relationship-type will not be installed (see message DBD5501).

User Response: A value must be specified for the indicated attribute before installation of the category or reltype may be completed.

**DBD5512 E NO ATTRIBUTE INFORMATION
 SPECIFIED FOR type**

Explanation: No attribute data has been specified in the definition for the subject category, relationship-type, or attribute-type. The variable "type" will be one of the keywords CATEGORY, RELTYPE or ATTRTYPE, as appropriate.

System Action: Installation execution is continued, but the subject category or relationship-type will not be installed (see message DBD5501).

User Response: The relevant category, relationship-type, or attribute-type definition must be completed before the category or relationship-type may be installed.

**DBD5513 E NO KEYWORD SPECIFIED FOR
 ATTRIBUTE**

Explanation: No identifying KEYWORD has been specified in the relationship between a subject category or relationship-type definition and the definition for one of the attribute-types associated with that category or reltype.

System Action: Installation execution is continued, but the subject category or relationship-type will not be installed (see message DBD5501).

User Response: A unique KEYWORD must be specified for each attribute associated with a subject category or relationship-type.

**DBD5514 E ATTRIBUTE KEYWORD keyword
 NOT UNIQUE**

Explanation: Each of the attributes associated with a subject category or relationship-type must be assigned a

unique identifying keyword. Two (or more) of the attributes associated with the category or relationship-type have been assigned the same keyword ("keyword").

System Action: Installation execution is continued, but the subject category or relationship-type will not be installed (see message DBD5501).

User Response: The KEYWORDS assigned to one (or more) of the attributes associated with the category or relationship-type definition must be changed in order to make them unique.

**DBD5515 E 'REPEAT' FACTOR FOR
 ATTRIBUTE keyword INVALID**

Explanation: An attribute may "repeat" between 1 and 99 times with respect to the subjects in a given category (or the instances of a given relationship-type). The REPEAT parameter on the relationship between the subject category or relationship-type definition and the definition for the attribute-type identified by the keyword "keyword" lies outside the valid 1-99 range.

System Action: REP=1 is assumed for the attribute, and installation execution is continued. The subject category or relationship-type will not however be installed (see message DBD5501).

User Response: A valid REPEAT parameter value must be specified on the category/attribute or reltype/attribute relationship.

**DBD5517 E VALUE FOR attribute
 ATTRIBUTE IS INVALID**

Explanation: During the installation of a subject category, relationship-type, or attribute-type definition, the value that has been specified for the indicated attribute ("attribute") within that definition has been found to be invalid.

System Action: Installation execution for the subject category or relationship-type definition is terminated.

User Response: If the installation has executed no programs that write data directly into the Dictionary "Extensibility" DL/I data base, the IBM representative should be consulted. If data has, in fact, been written directly into the Dictionary DL/I data bases, the value for the indicated attribute should be corrected within the category, relationship-type, or attribute definition.

**DBD5518 E ATTRIBUTE KEYWORD keyword
RESERVED**

Explanation: Each of the attributes associated with a subject category or relationship-type must be assigned a unique identifying keyword. The following is a list of reserved keywords which are not available as attribute keywords: DESC, TEXT, TEXTSEQ, USER1, USER2, USER3, USER4, USER5, FROM, TO, EXISTING.

System Action: Installation execution for the subject category or relationship-type definition is terminated.

User Response: The KEYWORDS assigned to the attribute associated with the category or relationship-type definition must be changed to a nonreserved keyword.

DBD5519 E ASSUMED VALUE: value

Explanation: The indicated value was assumed so error checking might continue. The assumption is internal to install execution and does not alter the value stored in the Dictionary.

System Action: See previously issued message.

User Response: See previously issued message.

**DBD5520 E INSUFFICIENT REGION
STORAGE TO PROCESS
ATTRIBUTES**

Explanation: The maximum number of different attributes (instances) that may be specified with respect to the subjects in any given subject category (or the instances of any given relationship-type) is limited by the amount of region storage available during installation execution. Not enough storage was available to build the attribute table.

System Action: Installation execution for the subject category or relationship-type is terminated.

User Response: The size of the Dictionary region should be increased for successful installation execution.

**DBD5521 E ATTRIBUTE DATA EXCEEDS
SEGMENT STORAGE MAXIMUM**

Explanation: The attribute data that may be stored with respect to the individual subjects in a given subject category (or the individual instances of a relationship-type) are formatted in the Dictionary "Extensibility" DL/I data base according to an algorithm outlined in the Program Logic Material (PLM) for the Dictionary product. The

data for the attributes that have been specified with respect to the subject category or relationship-type definition would exceed the storage available under the formatting algorithm.

System Action: Installation execution is continued, but the subject category or relationship-type will not be installed (see message DBD5501). Note that the "FORMAT" report that may (if requested) follow the installation execution report may contain errors.

User Response: Either the set of attributes associated with the category or relationship-type definition must be reduced, or the attributes may be rearranged (reordered) within the set so as to provide for more efficient (compact) storage of the attribute data within the Dictionary "Extensibility" data base.

**DBD5522 E VALIDATION DATA EXCEEDS
SEGMENT STORAGE MAXIMUM**

Explanation: The validation data for the attributes that may be specified with respect to the subjects in a given category (or the instances of a given relationship-type) are formatted in the Dictionary Extensibility "control information" according to an algorithm outlined in the Program Logic Material (PLM) for the Release 3 Dictionary product. The validation data for the attributes associated with the subject category or relationship-type definition would exceed the storage available under the formatting algorithm.

System Action: Installation execution is continued, but the subject category or relationship-type will not be installed (see message DBD5501).

User Response: The validation specifications (VALLIST or VALRNG) for one or more of the attribute-types associated with the subject category or relationship-type definition must be shortened, or be replaced by a validation routine.

**DBD5523 E INSUFFICIENT REGION
STORAGE TO PROCESS
VALIDATION SPECIFICATIONS**

Explanation: The maximum number of validation entries (that is, routine names, list values, range bounds) that may be specified with respect to the subjects in any given subject category (or the instances of any given relationship-type) is limited by the amount of region storage available during installation execution. Not enough storage was available to build the validation table.

System Action: Installation execution for the subject category or relationship-type is terminated.

User Response: The size of the Dictionary region should be increased for successful installation execution.

DBD5530 E type SPECIFIED AS IF INSTALLED

Explanation: The user has entered an INSTALL command requesting the installation of a subject category or relationship-type definition that already has "Installed" status. The variable "type" will be either of the keywords CATEGORY or RELTYPE, according to whether the message is issued in response to an INSTALL CATEGORY or INSTALL RELTYPE command.

System Action: Execution of the INSTALL command is terminated.

User Response: Check that the correct category or reltype subject name was specified in the INSTALL command.

DBD5531 E type name NOT DEFINED IN DICTIONARY

Explanation: The user has entered an INSTALL command requesting the installation of a subject category or relationship-type that is not defined in the Dictionary. The variable "type" will be either of the keywords CATEGORY or RELTYPE. The Dictionary subject name for the category or reltype definition specified in the INSTALL command will appear in the message in place of the variable "name".

System Action: Execution of the INSTALL command is terminated.

User Response: Check to be sure that the correct category or relationship-type subject name was specified in the INSTALL command.

DBD5532 E type OF SAME NAME ALREADY INSTALLED

Explanation: There already exists, in the Dictionary, an installed subject category or relationship-type or definition having the same user-name as the category or reltype specified in the INSTALL command -- or an installed attribute-type definition having the same user-name and occurrence number as the indicated attribute. The variable "type" will be one of the keywords CATEGORY, RELTYPE or ATTRTYPE, as appropriate.

System Action: Installation execution is continued, but the subject category or relationship-type will not be installed (see message DBD5501).

User Response: The user-name for the subject category, relationship-type, or attribute-type must be changed before the category or reltype may be installed.

DBD5533 E ALIAS NAME name FOR type ALREADY IN USE

Explanation: One of the "alias" names for the subject category or relationship-type specified in the INSTALL command is already in use with respect to another installed category or reltype. The invalid alias name will be displayed in the message in place of the "name" variable. The variable "type" will be either of the keywords CATEGORY or RELTYPE, according to whether the message is issued in response to an INSTALL CATEGORY or INSTALL RELTYPE command.

System Action: Installation execution is continued, but the subject category or relationship-type will not be installed (see message DBD5501).

User Response: The indicated "alias" name must be changed.

DBD5541 E MINNAME VALUE GREATER THAN MAXNAME VALUE

Explanation: The MINNAME (minimum user-name length) value that has been specified in the definition for a subject category is greater than the MAXNAME (maximum user-name length) value.

System Action: Installation execution is continued, but the subject category will not be installed (see message DBD5501).

User Response: Either the MINNAME or MAXNAME value must be corrected in the category definition.

DBD5542 E SBJCODE ALREADY IN USE FOR CATEGORY category

Explanation: The subject-code value (SBJCODE) specified in the definition for the subject category has already been used in conjunction with another installed subject category. The name of the installed subject category appears in the message in place of the variable "category".

System Action: Installation execution is continued, but the subject category will not be installed (see message DBD5501).

User Response: Another SBJCODE value must be selected for the subject category before the category may be installed.

DBD5543 I MINNAME VALUE DEFAULTED TO 1

Explanation: No minimum user-name length specification (MINNAME) has been provided in the definition for the subject category, and the default MINNAME=1 value has been assumed.

System Action: Installation execution for the subject category is continued.

User Response: If the INSTALL command has been entered with the UPDATE=NO option, and a MINNAME specification is required for the subject category, the specification must be added into the category definition before the INSTALL command is executed with the UPDATE=YES option.

DBD5544 I MAXNAME VALUE DEFAULTED TO 31

Explanation: No maximum user-name length specification (MAXNAME) has been provided in the subject category definition, and the default MAXNAME=31 value has been assumed.

System Action: Installation execution for the subject category is continued.

User Response: If the INSTALL command has been entered with the UPDATE=NO option, and a MAXNAME specification is required for the subject category, the specification must be added to the category definition before the INSTALL command is executed with the UPDATE=YES option.

DBD5545 I NAMETYPE VALUE DEFAULTED TO 'N'

Explanation: No user-name data-type rule (NAMETYPE) has been specified in the definition for the subject category, and the default NAMETYPE=N value has been assumed.

System Action: Installation execution for the subject category is continued.

User Response: If the INSTALL command has been entered with the UPDATE=NO option, and a NAMETYPE specification for the category is required, the specification must be added to the category definition before the INSTALL command is executed with the UPDATE=YES option.

DBD5551 E keyword NOT VALID AS A RELATIONSHIP-TYPE name

Explanation: One of the reserved keywords "WITH", "CONTAINS", "REDEFINES", "RENAMES" or "RENAMES_THRU" has been specified as either the "forward" or "inverse" name for an Extensibility relationship-type. The variable "name" in the message will be either FORNAME or INVNAME, according to whether it is the "forward" or "inverse" name for the relationship-type that has been

assigned the invalid keyword.

System Action: Installation execution is continued, but the relationship-type will not be installed (see message DBD5501).

User Response: Another relationship keyword (name) must be selected for the relationship-type.

DBD5552 E left/right CATEGORY category NOT INSTALLED

Explanation: The user has entered an INSTALL command with the UPDATE=YES option for an Extensibility relationship-type, and either the "left-" or "right-hand" category for the relationship-type has not itself been installed. The variable "left/right" in the message will be replaced by either of the keywords LEFT-HAND or RIGHT-HAND, as appropriate. The user-name for the uninstalled category will appear in the message in place of the variable "category".

System Action: Installation execution is continued, but the relationship-type will not be installed (see message DBD5501).

User Response: The indicated subject category must be installed before the relationship-type may be installed.

DBD5553 E RELATIONSHIP NOT UNIQUE FOR CATEGORY category

Explanation: Either the "forward" or "inverse" name (FORNAME or INVNAME) for the relationship-type is such that the relationship would not be unique for subjects in the indicated category ("category").

System Action: Installation execution is continued, but the relationship-type will not be installed (see message DBD5501).

User Response: Either the "forward" or "inverse" name (FORNAME or INVNAME) for the relationship-type must be changed so that the relationship will be unique for the indicated category.

DBD5555 E SEQATTR SPECIFICATION INVALID FOR UNSEQUENCED RELTYPE

Explanation: A sequence attribute (SEQATTR) specification has been provided in the definition for an unsequenced (SEQOPT=N) relationship-type.

System Action: Installation execution is continued, but the relationship-type will not be installed (see message DBD5501).

User Response: Either the SEQOPT attribute in the relationship-type definition must be corrected, or the SEQATTR specification deleted.

**DBD5556 E SEQATTR SPECIFICATION
REQUIRED FOR SEQUENCED
RELTYPE**

Explanation: No sequence attribute (SEQATTR) has been specified in the definition for a sequenced (SEQOPT=Y) relationship-type.

System Action: Installation execution is continued, but the relationship-type will not be installed (see message DBD5501).

User Response: Either a sequence attribute (SEQATTR) must be specified for the relationship-type, or the SEQOPT specification must be corrected.

**DBD5557 E SYMMETRIC RELTYPE WITHIN
ONE CATEGORY CANNOT BE
SEQUENCED OR DIRECTED**

Explanation: The Release 3 Extensibility facility will not support sequenced (SEQOPT=Y) relationship-types that are "symmetrical" (that is, where the "forward" and "inverse" names, FORNAME and INVNAME, are identical) where both the "left-" and "right-hand" categories (LCATNAME and RCATNAME) are the same.

System Action: Installation execution is continued, but the relationship-type will not be installed (see message DBD5501).

User Response: The definition for the relationship-type must be corrected.

**DBD5558 E SEQUENCE ATTRIBUTE
attribute MUST BE FIRST
ATTRIBUTE OF RELTYPE**

Explanation: The sequence attribute (SEQATTR) that has been specified for the relationship-type is not the first attribute associated with that attribute-type.

System Action: Installation execution is continued, but the relationship-type will not be installed (see message DBD5501).

User Response: The set of attributes associated with the relationship-type must be reordered so that the specified sequence attribute occurs first in the set.

**DBD5559 E DATATYPE FOR SEQUENCE
ATTRIBUTE INVALID**

Explanation: The sequence attribute (SEQATTR) for the relationship-type does not have one of the permissible

data-types. The data-type (DATATYPE) of the attribute that is to serve as the sequence attribute for an Extensibility relationship-type must be either "Z" (numeric), "A" (alphabetic), "C" (alphameric), or "N" ("name").

System Action: Installation execution is continued, but the relationship-type will not be installed (see message DBD5501).

User Response: Either the DATATYPE of the specified sequence attribute-type must be changed, or the definition for the relationship-type must be changed to specify a different sequence attribute.

**DBD5560 E DATAMAX FOR SEQUENCE
ATTRIBUTE IS GREATER THAN
8**

Explanation: The maximum data length (DATAMAX) of the specified sequence attribute (SEQATTR) for the relationship-type is invalid. The attribute-type that is specified as the sequence attribute for an Extensibility relationship-type must have DATAMAX<=8.

System Action: Installation execution is continued, but the relationship-type will not be installed (see message DBD5501).

User Response: The DATAMAX attribute for the sequence attribute-type must be corrected, or the definition for the relationship-type must be changed to specify a different sequence attribute.

**DBD5561 E BOTH CATEGORIES FOR
RELTYPE ARE
NON-EXTENSIBILITY**

Explanation: The Release 3 Extensibility facility will not support Extensibility relationship-types between pairs of "non-extensibility" subject categories. Both the "left-" and "right-hand" categories (LCATNAME and RCATNAME) specified in the definition for the relationship-type are "non-extensibility" categories.

System Action: Installation execution for the relationship-type is terminated.

User Response: The type of relationship desired by the user is not supported in the Release 3 Dictionary product.

**DBD5562 E RELATIONSHIP TO 'CONTROL
INFORMATION' CATEGORY
INVALID**

Explanation: The Release 3 Extensibility facility does not support Extensibility relationship-types where one (or both) of the categories is one of the special "control information" categories (that is, CATEGORY, RELTYPE or ATTRTYPE).

System Action: Installation execution for the relationship-type is terminated.

User Response: The type of relationship desired by the user is not supported by the Release 3 Dictionary product.

DBD5563 E SEQUENCE ATTRIBUTE attribute REPEATS

Explanation: The sequence attribute for a RELTYPE may not repeat.

System Action: Installation execution is continued, but the subject CATEGORY or RELATIONSHIP-TYPE will not be installed (see message DBD5501).

User Response: Correct the error, and reexecute INSTALL.

DBD5571 E DATAMIN VALUE GREATER THAN DATAMAX VALUE

Explanation: The DATAMIN (minimum data-length) value that has been specified in the definition for an attribute-type is greater than the DATAMAX (maximum data-length) value.

System Action: Installation execution is continued, but the subject category or relationship-type will not be installed (see message DBD5501). For "formatting" purposes, the DATAMAX value specified for the attribute-type will be assumed to be correct.

User Response: Either the DATAMIN or DATAMAX values in the attribute-type definition must be corrected before the category or reltype can be installed.

DBD5572 I DATAMIN DEFAULTED TO 1

Explanation: This informational message is issued during the installation execution for a subject category or relationship-type to indicate that no DATAMIN value has been specified for the preceding attribute, and the default DATAMIN=1 has been assumed.

System Action: None.

User Response: If the INSTALL command was entered with the UPDATE=NO option, and a DATAMIN of other than one is desired for the attribute, the attribute-type definition must be updated before the subject category or

relationship-type definition is installed with UPDATE=YES.

DBD5573 I NO VALIDATION SPECIFIED FOR ATTRIBUTE-TYPE

Explanation: This informational message is issued during the installation execution for a subject category or relationship-type to indicate that no validation has been specified for the preceding attribute.

System Action: None.

User Response: If the INSTALL command was entered with the UPDATE=NO option, and some validation for the attribute is desired, the attribute-type definition should be updated before the subject category or relationship-type definition is installed with the UPDATE=YES option.

DBD5575 E VALIDATION ROUTINE NOT SPECIFIED FOR VALIDATE=RTN

Explanation: Validation by routine (VALIDATE=RTN) has been specified in the definition for the attribute-type, and no validation routine (VALRTN) has been specified.

System Action: Installation execution is continued, but the subject category or relationship-type will not be installed (see message DBD5501).

User Response: Either the VALIDATE option or the VALRTN specification must be corrected in the attribute-type definition.

DBD5576 E DATAMAX MAY NOT EXCEED 24 FOR LIST/RANGE VALIDATION

Explanation: "List" or "Range" validation (VALIDATE=LIST|RNG) has been specified for the attribute-type, together with a DATAMAX (maximum data-length) value greater than 24. Attributes to be validated against a list or range must have a DATAMAX of 24 or fewer characters.

System Action: Installation execution is continued, but the subject category or relationship-type will not be installed (see message DBD5501).

User Response: Either the DATAMAX or the VALIDATE option must be changed in the attribute-type definition.

DBD5577 E DATATYPE INVALID FOR LIST/RANGE VALIDATION

Explanation: The data-type (DATATYPE) for an attribute whose values are to be validated against a list or a "range" (VALIDATE=LIST|RNG) may not be either "Q" ("Quoted-string") or "L" ("List").

System Action: Installation execution is continued, but the subject category or relationship-type will not be installed (see message DBD5501).

User Response: Either the DATATYPE or VALIDATE specification must be corrected in the attribute-type definition.

DBD5578 E VALRNL/H NOT SPECIFIED FOR VALIDATE=RNG

Explanation: Validation against a "range" (VALIDATE=RNG) has been specified for an attribute-type, but not both range specifications (VALRNL and/or a VALRNGH) have been provided.

System Action: Installation execution is continued, but the subject category or relationship-type will not be installed (see message DBD5501).

User Response: Either the VALIDATE option in the attribute-type definition must be changed, or both a VALRNL and a VALRNGH specification provided.

DBD5579 E VALLIST NOT SPECIFIED FOR VALIDATE=LIST

Explanation: Validation against a "list" (VALIDATE=LIST) has been specified in the definition for an attribute-type, and no list specification (VALLIST) has been provided.

System Action: Installation execution is continued, but the subject category or relationship-type will not be installed (see message DBD5501).

User Response: Either the VALIDATE option must be changed, or a VALLIST specification provided in the attribute-type definition.

DBD5580 E DATAMAX OR DATAMIN DOES NOT CONFORM TO DATATYPE

Explanation: For an Attribute-Type with DATATYPE of L or Q, the DATAMAX, and DATAMIN must be at least 3 to allow room for the delimiting characters.

System Action: The Category or Reltype is not installed.

User Response: Correct the error in the Attribute-Type definition.

DBD5581 E keyword VALUE ERROR IN VALIDATION LIST/RANGE SPECIFICATION

Explanation: Each of the values specified in the validation list (VALLIST) and/or validation range (VALRNL and VALRNGH) specification for an attribute-type must conform to

the DATAMIN, DATAMAX, and DATATYPE specifications for that attribute-type. The variable "keyword" in the message will be replaced by the keyword of the attribute in error; it is either VALRNL, VALRNGH, or VALLIST(xx), where xx is the instance of the VALLIST specification.

System Action: Installation execution is continued, but the subject category or relationship-type will not be installed (see message DBD5501).

User Response: The invalid values occurring in the VALLIST or VALRNL/H specification for the attribute-type must be corrected.

DBD5582 E VALIDATION RANGE (VALRNL/H) SPECIFICATION INVALID

Explanation: The "minimum" and "maximum" values specified in the validation range specification (VALRNL and VALRNGH) for an attribute-type must, in fact, define a valid "range". The "maximum" and "minimum" values specified in the VALRNL/H specification for the attribute-type are equal (in which case the one valid value should be specified in a validation list, VALLIST), or the "minimum" exceeds the "maximum."

System Action: Installation execution is continued, but the subject category or relationship-type will not be installed (see message DBD5501).

User Response: The validation range specification (VALRNL/H) in the attribute-type definition must be corrected.

DBD5589 S ERROR IN DEFINITION FOR INSTALLED type name

Explanation: Information vital to Extensibility execution is missing from the installed CATEGORY or RELTYPE definition. Specifically, the EXTATR segment is missing from the indicated definition.

System Action: Installation execution for the subject CATEGORY or RELTYPE is terminated.

User Response: The absence of this information indicates a serious problem. Notify your IBM representative.

GENERAL EXTENSIBILITY DISPLAY FORMS (DBD56NN T)

DBD5601 I 'EXPLAIN' TEXT NOT DEFINED

Explanation: The user has requested a display of the "Explanation" for an Extensibility subject category or relationship-type, and no "Explain" text has been provided by the installation for that category or reltype.

System Action: None.

User Response: The user might request that the installation "Dictionary Administration" function provide the "Explain" text for the subject category or relationship-type.

DBD5602 E NO ENTRY IN LIST HAS BEEN 'MARKED'

Explanation: The user has requested an action on a "list"-type display-form requiring that a specific entry in the list be identified by a "mark" (that is, an asterisk entered into the appropriate "Mark" column). Either no entry in the list was so marked, or some character other than an asterisk was used for the "mark" character.

System Action: Execution of the display-form is terminated.

User Response: The user should "mark" the desired entry with an asterisk and request the same action again.

DBD5603 E 'MARKED' ENTRY IS EMPTY OR INCOMPLETE

Explanation: The user has requested an action on a "list"-type display-form requiring that a specific entry in the list be identified by a "mark" (that is, an asterisk entered into the appropriate "Mark" column). The entry in the list that was so marked by the user is either empty or incomplete.

System Action: Execution of the display-form is terminated.

User Response: Only complete, processed entries in a "list"-type display can be the object of an action request.

DBD5605 E SUBJECT category NOT DEFINED

Explanation: The user has accessed a display-form (for example, SUBJECT-RELATIONSHIPS) requiring that an entry already exist in the Dictionary for a particular subject. No entry, however, exists in the Dictionary for that subject at this point. Within the text of the message, "category" will be replaced by the name of the subject category in which the subject is supposed to reside.

System Action: Further execution of the display-form is terminated.

User Response: Before the display-form can be processed, an entry must exist in the Dictionary for the indicated subject. This will generally require returning to the Subject-Definition form to create that entry.

DBD5606 E REQUEST VIOLATES USER ACCESS AUTHORIZATION

Explanation: The user has entered data or requested an action on a display-form that would result in the retrieval and display of data regarding a subject for which he has no "view" authorization.

System Action: The data in question is not retrieved for display. Further execution is terminated.

User Response: The user may not request, through the display-forms, access to data to which he has no authorization under the Dictionary basic security facility.

DBD5607 E SUBJECT CATEGORY 'category' DOES NOT EXIST

Explanation: In a field (on a display form) in which the name for a subject category is to be specified, the user has entered a name which does not, in fact, correspond to that of any of the subject categories currently existing in the Dictionary.

Within the text of the message, "category" will be replaced by the erroneous category name specified by the user.

System Action: Execution of the display form is terminated.

User Response: In some situations, the user may simply overwrite the erroneous category name entry with a valid category name, and again request the desired action.

In the specific case of the selected SUBJECT-RELATIONSHIPS form, where the user has accessed the form specifying an erroneous category name, the 9-RTN or 10-HDR action must first be invoked to return to the form from which the SUBJECT-RELATIONSHIPS form was accessed, then the form may be reaccessed by specifying a valid category name.

DBD5610 S ERROR AT POINT point-id IN MODULE module

Explanation: This is a general error message that is issued by Dictionary modules when they detect an error that prevents further execution. Within the text of the message, "module" is the name of the module in which the error was encountered, while "point-id" is a code identifying the point in that

module where the error condition was detected.

System Action: Further execution is terminated.

User Response: This error should not occur. The user should record the conditions under which the message was issued, and contact his IBM representative for diagnosis and correction of the problem giving rise to the error condition.

DBD5620 I SUBJECT category NOT DEFINED IN DICTIONARY

Explanation: This informational message is issued to indicate that no entry exists in the Dictionary for the subject requested by the user on the HEADER form. Within the text of the message, "category" will be the name of the Extensibility subject category in which the subject is to reside.

System Action: None.

User Response: To have an entry created in the Dictionary for the specified subject, the user may add additional information into the display-form, and request the "PROC" action.

DBD5621 E SUBJECT category DEFINITION DOES NOT EXIST

Explanation: The user has attempted to invoke the DELETE action on the EXTENSIBILITY SUBJECT form, and no definition currently exists in the Dictionary for the subject displayed thereon.

System Action: Execution of the display-form is terminated.

User Response: A subject definition that does not exist in the Dictionary cannot, of course, be deleted.

DBD5622 E INVALID 'RELATIONSHIPS' REQUEST

Explanation: When the definition for an Extensibility "Control Information" subject (that is, a CATEGORY, RELTYPE or ATTRTYPE subject) is displayed on the EXTENSIBILITY SUBJECT form, only certain special "options" are valid for the 5-RELATIONSHIPS action. The option specified by the user (in the field following the 5-RELATIONSHIPS action keyword) was not one of the valid options.

System Action: Further execution of the display-form is terminated.

User Response: The user should consult the display forms contained in the DB/DC Data Dictionary Administration and Customization Guide to find out

which specific options are valid for the 5-RELATIONSHIPS action when the definition for a "Control Information" subject is displayed on the EXTENSIBILITY SUBJECT form.

DBD5630 E category NOT DEFINED IN DICTIONARY

Explanation: The user has accessed the CATEGORY/RELTYPE-ATTRIBUTES display-form without first causing an entry for the subject category or relationship-type to be added to the Dictionary. Within the text of the message, "category" will be either "CATEGORY" or "RELTYPE", as appropriate.

System Action: Further execution of the display-form is terminated.

User Response: If the user wishes to create an entry in the Dictionary for the new subject category or relationship-type, he must "return" (RTN action) to the EXTENSIBILITY SUBJECT form and request the "PROC" action.

DBD5631 E ATTRTYPE SUBJECT NAME MAY NOT BE CHANGED

Explanation: The user has attempted to change or erase the subject name for an attribute-type in an entry in the attribute table in the CATEGORY/RELTYPE-ATTRIBUTES display-form. These attribute-type subject names may not be changed.

System Action: The original attribute-type subject name is restored to the display form, and execution is terminated.

User Response: If the user wishes to change one of the attribute-types to which the category or relationship-type definition is related, he must first delete the relationship to the old attribute-type, and then add a relationship to the new attribute-type.

DBD5632 E SEQ VALUE MUST BE SPECIFIED FOR ATTRIBUTE

Explanation: The user has entered the subject name for an attribute-type into a line in the attributes table of the CATEGORY/RELTYPE-ATTRIBUTES display-form, but has failed to specify a SEQ value for the relationship. As outlined in the DB/DC Data Dictionary Administration and Customization Guide, a unique SEQ value must be associated with each of the relationships between a subject category or relationship-type definition and its associated attribute-types.

System Action: Further execution of the display-form is terminated.

User Response: If the user wishes to create a relationship between the category or relationship-type definition and the specified attribute-type, he must specify a unique SEQ value for the relationship.

DBD5633 E ATTRTYPE SUBJECT NAME NOT SPECIFIED

Explanation: The user has entered a partial entry in the attributes table of the CATEGORY/RELTYPE-ATTRIBUTES display-form, but has failed to provide an attribute-type subject name within that entry.

System Action: Further processing of the display-form is terminated.

User Response: The user should either complete the entry with an attribute-type subject name (in which case a relationship to that attribute-type will be established when the form is PROCessed), or erase the data in the fields comprising the partial entry using the Erase EOF key on his 3277 terminal.

CATEGORY-RELTYPES OR RELATIONSHIPS MESSAGES (5640-49)

DBD5640 S CATEGORY NOT INSTALLED

Explanation: The user has managed to access either the CATEGORY-RELTYPES or CATEGORY-RELATIONSHIPS display-form without the subject category being "installed" in the Dictionary.

System Action: Further execution of the form is terminated.

User Response: This situation should not occur. The user should record the conditions under which this message was issued, and contact his IBM representative for diagnosis and correction of the error.

DBD5661 E TO-CATEGORY NAME MAY NOT BE MODIFIED

Explanation: The user has attempted to erase or change the "to" CATEGORY name in a relationship entry on the Unselected SUBJECT-RELATIONSHIPS form.

System Action: The affected "to" CATEGORY name is restored to the display-form, and further execution of the form terminated.

User Response: These names may not be changed or erased.

DBD5662 E RELATIONSHIP SPECIFICATION INCOMPLETE

Explanation: The user has entered a partial or incomplete relationship entry in the relationships table of the SUBJECT-RELATIONSHIPS form. On the Selected SUBJECT-RELATIONSHIPS form, a complete relationship entry consists of:

- The RELATIONSHIP keyword identifying the type of relationship to be established,
- Subject name (SUBJECT2-NAME, CODE, OCC, and STAT) identifying the subject to be related, and
- The "mode" code (specified in the "MARK" column) indicating whether the subject to be related is "new" or "old."

On the Unselected SUBJECT-RELATIONSHIPS form, a complete relationship entry consists of the same data, plus the name of the subject category (CATEGORY) in which the related subject resides (or is to reside). On either form, a SEQUENCE value must be specified in an entry if (and only if) the type of relationship to be established has been defined to be "sequenced".

System Action: Further execution of the display-form is terminated.

User Response: The user must complete the relationship entry with all required values before the form can be processed.

DBD5663 E RELATED SUBJECT NAME MAY NOT BE MODIFIED

Explanation: The user has attempted to change or erase the related subject name in a relationship entry on the SUBJECT-RELATIONSHIPS form.

System Action: The original related subject name is restored to the form, and further execution of the form terminated.

User Response: The related subject name may not be changed or erased. If the user wishes to change the subject that is the "target" of the relationship, he must first delete the existing relationship, and then create a new relationship to the new "target" subject. If the user wishes to delete the relationship, he may do so by crossing-out the RELATIONSHIP keyword (rather than the related subject name) with "-" signs.

DBD5664 E RELATIONSHIP (KEYWORD) MAY NOT BE MODIFIED

Explanation: The user has attempted to change the relationship keyword (RELATIONSHIP) in one of the relationship entries on the SUBJECT-RELATIONSHIPS form.

System Action: The original RELATIONSHIP keyword is restored to the form, and further execution of the form terminated.

User Response: The RELATIONSHIP keyword in a relationship entry may not be modified. If the user wishes to establish a different type of relationship to the related subject, he must first delete the existing relationship, and then add an entry for the new relationship.

DBD5665 E SPECIFIED category SUBJECT ALREADY EXISTS

Explanation: In a new relationship entry in the SUBJECT-RELATIONSHIPS or CATEGORY/RELTYPE-ATTRIBUTES form, the user has indicated (via the "mode" code entered into the "MARK", "MK", or "MODE" column) that the related subject specified in that entry is "New." A subject with the specified subject name already exists, however, in the specified subject category ("category").

System Action: Further execution of the form is terminated.

User Response: The user should check that he has specified the proper related subject name in the relationship entry.

DBD5666 E SPECIFIED category SUBJECT DOES NOT EXIST

Explanation: In a new relationship entry in the SUBJECT-RELATIONSHIPS or CATEGORY/RELTYPE-ATTRIBUTES form, the user has indicated (via a "mode" code entered into the "MARK", "MK" or "MODE" column) that the related subject specified in that entry is "Old." No subject with the specified subject name exists, however, in the specified subject category ("category").

System Action: Further execution of the display-form is terminated.

User Response: The user should check that he has specified the proper related subject name in the relationship entry.

DBD5667 E SEQUENCE ATTRIBUTE VALUE MAY NOT BE SPECIFIED

Explanation: The user has attempted to add a SEQUENCE value to an existing (that is, previously processed) relationship entry on the SUBJECT-RELATIONSHIPS form. Since the

relationship entry could not have been processed successfully if a SEQUENCE value was in fact required with respect to the relationship represented by that entry, later addition of a SEQUENCE value would be invalid.

System Action: Further execution of the form is terminated.

User Response: The value that the user has entered into the SEQUENCE field of that entry should be erased using the Erase EOF key of the 3277 terminal.

DBD5668 E SEQUENCE ATTRIBUTE VALUE MAY NOT BE DELETED

Explanation: The user has attempted to erase the SEQUENCE value in an existing (that is, previously processed) relationship entry on the SUBJECT-RELATIONSHIPS form. Since the relationship entry could not have been processed successfully unless the SEQUENCE value was indeed required for the type of relationship represented by that entry, deletion of the value would be invalid.

System Action: The original SEQUENCE value is restored to the form, and further execution of the form is terminated.

User Response: None.

DBD5669 E category NOT AN 'EXTENSIBILITY' CATEGORY

Explanation: The user has attempted, via an entry on the Unselected SUBJECT-RELATIONSHIPS form, to establish a relationship between the principal subject and a subject in a non-Extensibility subject category ("category"). Only relationships to subjects residing in Extensibility subject categories may be established via entries on the Unselected SUBJECT-RELATIONSHIPS form.

System Action: Further execution of the display-form is terminated.

User Response: The user must correct the erroneous entry or remove it from the display-form before any subsequent entries may be processed.

DBD5670 E RELATED category SUBJECT NAME INVALID

Explanation: On either of the SUBJECT-RELATIONSHIPS forms, the user has specified (in a new entry) a subject name containing an invalid occurrence number qualifier.

System Action: Further execution of the form is terminated.

User Response: The user must correct the erroneous occurrence number before any subsequent entries in the form may be processed.

DBD5670 E RELATED category SUBJECT NAME INVALID

Explanation: On either of the SUBJECT-RELATIONSHIPS forms, the user has specified (in a new entry) a subject name containing an invalid occurrence number qualifier.

System Action: Further execution of the form is terminated.

User Response: The user must correct the erroneous occurrence number before any subsequent entries in the form may be processed.

DBD5681 E SEQUENCE ATTRIBUTE VALUE MAY NOT BE DELETED

Explanation: The user has attempted to delete the (SEQUENCE) attribute value displayed on the RELATIONSHIP-DATA form.

System Action: The original (SEQUENCE) attribute value is restored to the form, and further execution of the form terminated.

User Response: Since the (SEQUENCE) attribute value is only displayed on the RELATIONSHIP-DATA form when the type of relationship that is displayed thereon does, in fact, require a (SEQUENCE) value (that is, the relationship-type is "sequenced"), the (SEQUENCE) value cannot be deleted.

DBD5682 S RELATIONSHIP DOES NOT EXIST

Explanation: The user has accessed the RELATIONSHIP DATA form to edit/display the relationship data for a specific relationship (instance), and that relationship cannot be found in the Dictionary.

System Action: Further processing of the form is terminated.

User Response: Under normal circumstances, this condition should not occur (since the RELATIONSHIP DATA form is accessed from one of the SUBJECT-RELATIONSHIPS forms by "marking" the entry for an existing relationship instance). If, upon returning to the SUBJECT-RELATIONSHIPS form, the same relationship (instance) is displayed in the relationships table, contact your IBM representative.

In rare circumstances, this message may indicate that there has been concurrent user interference. That is to say, another user has deleted the relationship instance during the time

that the RELATIONSHIP DATA form was being generated for display.

SIGN ON (DBD60NN T)

DBD6001 E SIGN_ON ID (name) NOT RECOGNIZED OR BLANK.

Explanation: The ID entered for "Sign-On" from the HEADER form, SIGN_ON command, received from IMS/VS, or received from CICS cannot be found in the DDUSER subject category or blanks were entered for the ID.

System Action: If using display-forms, the HEADER form is redisplayed for another "Sign-On" attempt. If using batch or online command mode, subsequent commands are syntax checked until another SIGN_ON statement appears in the input stream.

User Response: Correctly specify the ID, or have the security coordinator at your location create an ID for your use.

DBD6002 E INVALID OR BLANK PASSWORD FOR SIGN_ON ID (id)

Explanation: The specified "sign-on" id has been found in the DDUSER subject category, but the password specified does not agree with the id given or was entered as blank at "sign-on" time or is stored as blank in the subject's password attribute.

System Action: If using display-forms, the Header form is redisplayed for another "Sign-On" attempt. If using batch or online command mode, subsequent commands are syntax checked until another SIGN_ON statement appears in the input stream.

User Response: Correctly specify the password, or have the security coordinator at your location update the password for your ID.

DBD6004 E STATUS CODE (status-code) NOT AUTHORIZED FOR operation

Explanation: Execution of a subject with the specified status code is not authorized for the operation specified. The operation may be "update" or "view."

System Action: Execution is terminated.

User Response: Select a status code which is authorized or only attempt authorized operations. You may need to contact the security coordinator at your installation to have your authorization changed.

**DBD6005 I SIGN-ON COMPLETED
SUCCESSFULLY -- DICTIONARY
SECURITY enabled/disabled**

Explanation: The ID, PASSWORD, and Access Description records for this user have been validated, and this user's Dictionary access authorization is established. If the Dictionary is using IMS or CICS "Sign_on" ID's, passwords are not checked.

System Action: This message will be output after a SIGN_ON command has been entered in batch or online command mode. It will also be displayed on the HEADER form after an ID and PASSWORD are processed. If running online without display forms and IMS/VS is being used, this message will be output along with the other messages associated with the first Dictionary command entered.

User Response: None required.

**DBD6006 E INVALID OPERAND FOR ACCESS
DESCRIPTION KEYWORD
(keyword)**

Explanation: An error was detected while processing an access description keyword:

- An unrecognizable value (not YES or NO) was specified for the VALCMND keyword.
- Specified status code is not in the range A-T, *, or 0-9.
- A parenthesis is missing on the VIEWSTAT or UPDTSTAT specification.
- The generic operand of the UPDTSTAT specification was not ALL, TEST, or NONE.
- The generic operand of the viewstat specification was not ALL or TEST.
- The generic operand of the VALCAT specification was not: ALL, STANDARD, ALL-FORMAT, STANDARD-FORMAT, or FORMAT.
- The END statement or some access description statement contained extraneous nonblank characters.

System Action: Execution continues with the next access description keyword. "Sign-on" will be refused after execution of the access description is complete.

User Response: Have the security coordinator at your location correct the DDUSER access description records.

**DBD6007 W POTENTIAL DICTIONARY
SECURITY VIOLATION--BY
batch/online USER (id) [at**

terminal 'lterm']

Explanation: A user has tried to enter an ID/PASSWORD a number of times without success. The exact number is contained in SECTHOLD, a security default value.

System Action: The count of attempts is reset to zero. The Dictionary program waits for a valid ID/PASSWORD in online mode. In batch mode, all commands associated with the failing SIGN_ON are flushed until another valid SIGN_ON command is encountered.

User Response: Have the security coordinator at your location instruct the user on proper Dictionary sign-on procedures. It is possible the user has been given the wrong ID/PASSWORD combination or an expected ID/PASSWORD change has not been completed.

**DBD6008 E CATEGORY SELECTED IS NOT
AVAILABLE FOR USE.**

Explanation: A subject category was selected on the HEADER form or specified in a command that this user is unauthorized to use.

System Action: The Dictionary program waits for another category selection on the HEADER form or in batch mode reads the next command from input stream.

User Response: Verify that this category is the proper one to use. In Online mode, select another category. In batch, correct the failing command and resubmit the job. Or, you may need to contact the security coordinator at your installation to have your authorization changed.

**DBD6009 I OUTPUT WILL BE LIMITED TO
DDUSER'S VIEW STATUS
CODES.**

Explanation: A GLOSSARY or SCAN report is being performed. The status codes, contained in this user's VIEWSTAT specification, are being used to control what may be seen by this user.

System Action: The Dictionary program applies the VIEWSTAT security check.

User Response: None required.

**DBD6010 W ACCESS DESCRIPTION
TERMINATED WITHOUT AN
'END' STATEMENT**

Explanation: The access descriptions for a user are stored in User Data segments associated with a DDUSER subject in the DDUSER category. This message is issued when a physical end of this user's access description is reached before encountering an "END" statement.

System Action: Defaults are assumed for keywords not yet processed. If no errors have been detected to this point, this "sign-on" request will be honored.

User Response: Have the security coordinator at your location correct the DDUSER access description records.

DBD6011 E ERROR TYPE (code) DETECTED IN ENTRY (n) OF CATEGORY RECORD.

Explanation: Error detected while processing a formatted category record during SIGN_ON command execution. The error codes are:

- The category number is not in the range 1-255,
- The separator code is not a dash,
- The category name is missing or not left justified,
- The category name does not begin with an alphabetic character,
- Extraneous nonblank characters exist.

System Action: Execution continues with the next category field in the record or with the next access description record. "Sign-on" will be refused when access description processing is complete.

User Response: Have the security coordinator at your location correct the DDUSER access description records. Correct the category record in error.

DBD6012 E DUPLICATE KEYWORD (keyword) DETECTED IN DDUSER'S ACCESS DESCRIPTION

Explanation: The keyword was specified multiple times in the DDUSER's access description.

System Action: Processing continues with the next access description keyword. "Sign-on" will be refused when the processing of this access description is complete.

User Response: Have the security coordinator at your location correct the DDUSER access description records. Delete the duplicate access description keyword.

DBD6013 E DL/I ERROR ON DATABASE database name STATUS WAS status code LAST SEGMENT WAS segment name

Explanation: An unexpected DL/I return code was received while performing an operation against one of the Dictionary data bases.

System Action: SIGN_ON execution is terminated and the SIGN_ON will fail. If the error occurred while constructing the HEADER form at "Sign-On" time, the form will be displayed along with this message. If the error occurred when returning to the HEADER form or when doing a HEADER paging operation, the page that was read before the error containing the data will be displayed.

User Response: Take whatever steps are necessary to correct the condition responsible for the error return code.

DBD6014 E EXPECTED CATEGORY RECORDS ARE MISSING.

Explanation: A VALCAT specification of FORMAT, ALL-FORMAT, or STANDARD-FORMAT was not followed by one or more formatted category records.

System Action: Execution continues with the next keyword. "Sign-On" will be refused when the execution of this user's access description is complete.

User Response: Have the security coordinator at your location correct the DDUSER access description records. Correct the VALCAT specification or add formatted category records.

DBD6015 E DICTIONARY SECURITY REQUIRES A SIGN_ON COMMAND.

Explanation: The first command encountered in a batch job stream or online session was not a SIGN_ON command.

System Action: The command is flushed and all remaining commands will be flushed until a valid SIGN_ON command appears.

User Response: Insert a SIGN_ON command at beginning of job stream.

DBD6016 W UNEXPECTED ACCESS DESCRIPTION RECORD.

Explanation: An access description record was encountered that is either not expected or completely unrecognized. This message could be caused by:

- Formatted category records following a VALCAT=ALL or STANDARD specification
- A blank line
- An otherwise valid access description record that does not start in character position one
- An invalid access description record.

System Action: The record is skipped, and execution continues with the next record. If this is the only error encountered, "Sign-On" will be allowed.

User Response: Have the security coordinator at your location correct the DDUSER access description records.

DBD6017 E IMS/CICS SUPPLIED USERID (id) NOT RECOGNIZED.

Explanation: When the Dictionary installation default, SECIMS=YES, is specified, and interactive display forms or online functional linkedits are being used, the Dictionary expects IMS/VS or CICS to supply a userid. This message is issued when that ID is not present in the Dictionary DDUSER category. IMS/VS supplies the ID in the I/O PCB control block. CICS supplies the ID in the new SPA for the transaction.

System Action: The Dictionary terminates execution and will not allow "Sign-On" data to be entered directly to the Dictionary.

User Response: Signon to IMS/VS or CICS with an ID recognizable both to IMS/VS or CICS and the Dictionary. If this is not possible, have the security coordinator at your location make the necessary corrections. These corrections may involve:

- Adding the missing ID to the DDUSER category
- Adding the missing ID to the IMS/CICS security tables

DBD6018 E SECURITY HEADER FORM IS REQUIRED.

Explanation: Dictionary security is being used (SEC_DICT=YES) and a nonsecurity HEADER form was used to start the interactive display session.

System Action: The Dictionary terminates execution.

User Response: Use the required security HEADER form to start the interactive display session.

DBD6019 E NON-SECURITY HEADER FORM IS REQUIRED.

Explanation: Dictionary security is not being used (SEC_DICT=NO), and the security HEADER form was used to start the interactive display session.

System Action: The Dictionary terminates execution.

User Response: Use the required nonsecurity HEADER form to start the interactive display session.

DBD6020 E SIGN_ON IS NOT ALLOWED FROM COMMAND FORM

Explanation: The SIGN_ON command can only be used in batch or with online Dictionary command entry linkedits.

System Action: The command is rejected.

User Response: Enter some other command or return to the HEADER Form.

DBD6021 W A SIGN_ON IS ALREADY IN EFFECT--REQUEST IGNORED.

Explanation:

- This message can only occur during a Dictionary online session. In online mode, using functional linkedits for command entry, only one SIGN_ON is allowed. An attempt to change SIGN_ON IDs during a Dictionary interactive session will be rejected with the above message.
- In online mode, an IMS/VS or CICS SIGN_ON may be required prior to invoking the Dictionary. This environment is described to the Dictionary with an installation default that forces the Dictionary to do an automatic SIGN_ON with this IMS/VS or CICS user ID. When this environment is in effect at your installation, a manually entered user ID will be rejected when using online functional linkedits for command entry.

System Action: The Dictionary flushes the command and waits for another user action.

User Response: The user can terminate this Dictionary session with a /EXIT command and then reinvoke the Dictionary and SIGN_ON. If an IMS/VS or CICS user ID is being used by the Dictionary, that ID must also be changed before reinvoking the Dictionary.

DBD6022 E INCOMPLETE ACCESS DESCRIPTION WAS ENCOUNTERED

Explanation: The access description records are stored in User Data segments associated with subjects in the DDUSER Dictionary category. The possible reasons for this error message follow:

- Both UPDTSTAT and VIEWSTAT specifications are missing. At least one of these keywords must be specified.
- A specification of UPDTSTAT=NONE was made, and the VIEWSTAT specification is missing.

System Action: The Dictionary will refuse "Sign_On."

User Response: Have the security coordinator at your location correct the DDUSER access description records.

DBD6023 I SIGN_ON ATTEMPT FAILED

Explanation: One or more errors were encountered in this user's access description records. The specific error is identified in a previous message.

System Action: The Dictionary ignores the "Sign_On" attempt.

User Response: Have the security coordinator at your location correct the DDUSER access description records.

DBD6024 E CURRENT USER'S DDUSER SUBJECT HAS BEEN ALTERED.

Explanation: The access description for this Dictionary user has been modified since the user first "Signed_On". This could have been done by the current user with authorization access DDUSER subjects or by some other user who is applying maintenance to the DDUSER subjects. The message can only occur in the interactive display form environment.

System Action: The HEADER form is redisplayed with:

- A blank category section when the DDUSER subject could not be found or all category records have been deleted.
- Extraneous user-data segment occurrences when the category records have been relocated to some different position within the user-data segment.
- The interactive session is terminated.

User Response: Have the security coordinator at your location correct the DDUSER access description records.

DBD6025 E DDUSER IS NOT AUTHORIZED TO ACCESS COMMAND FORM

Explanation: The access description states that this user may not leave the interactive display form header and use the Dictionary command entry form.

System Action: The Dictionary waits for next action.

User Response: Choose another action on the form or contact the security coordinator to have your authorization changed.

DBD6026 I VIEW/UPDATE STATUS CODES ARE: code-string

Explanation: This message will contain a list of the status codes you are authorized to update or view. The message is issued twice, once for update and once for view. The message is displayed in batch, or when using the online functional link-edits.

System Action: The message is issued, and the Dictionary executes the next command.

User Response: None required.

DBD6027 E A RELATIONSHIP TO DDUSER SUBJECT IS NOT ALLOWED

Explanation: The DDUSER category cannot participate in relationships with any other subject categories.

System Action: The command or online request is rejected.

User Response: None.

DBD6028 E UNABLE TO LOCATE DDUSER'S ACCESS DESCRIPTION

Explanation: Each DDUSER subject has an attribute that is a pointer to where access description records are stored. If this attribute has not been specified, the default location is User Data segment number 1 and line number 1 within that user-data segment. If no access description records have been stored at this location, the above message is issued.

System Action: The request for Sign_on is rejected.

User Response: Have the security coordinator at your location correct the DDUSER access description records. The access description pointer must be corrected or one or more access description records entered for this DDUSER subject.

DBD6029 E EXCESSIVE ACCESS DESCRIPTION ERRORS -- PROCESSING TERMINATED

Explanation: The Sign_On processor issues this message after 'n' access description records have been processed, each having one or more error/warning messages issued on behalf of that record. The constant 'n' is currently set at 10 and can only be changed by reassembling the Sign_On processor.

System Action: The request for Sign_on is rejected.

User Response: Have the security coordinator at your location correct the DDUSER access description records.

**DBD6030 I ALL AVAILABLE CATEGORIES
ARE DISPLAYED -- ACTION
IGNORED**

Explanation: The TOP/DOWN actions on the security Header display form are not valid for a user whose security access description contains VALCAT=ALL

or VALCAT=STANDARD.

System Action: The request is ignored.

User Response: Continue with other actions available on the HEADER form.

INDEX

For further information about dictionary or DL/I concepts, you may refer also to the glossary in the DB/DC Data Dictionary General Information Manual or to the IMS/VIS Master Index and Glossary SH20-9085.

A

A command 188
(see also ADD command)
ACT field, display forms 29
action area in display forms 17, 29
actions on display forms 29-30
ADD command 188
ADD_RELATIONSHIP command 188-194
 for aliases 193-194
 for data base- or PCB-segment relationships 192
 for element-element relationships (not aliases) 189-190
 for installation-defined relationships 193
 for segment-element relationships 191
 for standard relationships 189
 restrictions on status of subjects 188-189
alias
 added with AR command 193-194
 categories in which accepted 3
 defined 2
 not deleted by DELETE command 209
 order of subjects in AR command 194
 restrictions on status 7
ALIAS action, display forms 29
ALIAS form 30-31
 contents 30
 description 30
 sample form 31
 use 31
aliases record
 on the Field Definition form 154
 on the Field Definition form for PL/I Data 162
 on the Segment Definition form 171
alternate action areas, display forms 17
anticipated attributes (see attributes)
AR command 188-194
(see also ADD_RELATIONSHIP command)
asterisk
 meaning in commands 182
 use with DOWN action on display forms 29
attributes 2

B

basic field attribute record

 on the Field Definition form 153
 on the Field Definition form for PL/I Data 161
basic segment attribute record 170
batch forms
 Field Definition form 150-158
 Field Definition form for PL/I Data 158-166
 Segment Definition form 167-173
 Text Data form 174-178
batch forms, general description 149-150
 contents 149
 defined 149
 forms available 149
 general use 149-150
batch input streams 147-148
batch-only commands 179
batch operation of dictionary
 in DL/I DOS/VIS 145-147
 in IMS/VIS 143-145
blank
 in commands 182-183
 in keyword=value expressions 183
 in SCAN command 236
blank lines
 on the RELATIONSHIP DATA form 99
 on the security HEADER form 24
brace, use of iv
bracket, use of iv
bulk entry of definitions
 with batch forms 149
 with commands
 (see COBOL_IN, DBD_IN, PLI_IN, and PSB_IN commands)

C

CAT-RELATIONSHIPS action
 on the selected SUBJECT-RELATIONSHIPS form 119
 on the unselected SUBJECT-RELATIONSHIPS form 121
CATEGORY-RELATIONSHIPS form 32-33
 contents 32
 description 32-33
 sample form 32
 specialized actions 33
 use 33
CBI command 196-198
(see also COBOL_IN command)
CHANGE_IN command 194-196
 using FROM and TO 195-196
CHANGE_NAME command 196
 restriction against change of status 196
CHANGE_RELATIONSHIP_DATA command 197-198
 using FROM and TO 195-196
character set
 in commands 182
 in SCAN command 232
 special rules for user names 3, 5
characteristics of subjects
(see attributes)
Checkpoint command (CKPT) 198
CI command 194-196
(see also CHANGE_IN command)

CICS/VS, online operations under 10-12
 CKPT command 198
 (see also Checkpoint command)
 CMNDMOD 9
 CN command 196
 (see also CHANGE_NAME command)
 COBOL attributes record 155-156
 COBOL attribute entries in AR command 190
 COBOL level 66 entries in AR command 191-192
 COBOL level 88 with USE parameter in AR command 190
 COBOL occurs clause record 156
 COBOL output with SO command 200
 COBOL value clause record 156
 COBOL_IN command 198-200
 entries created 198
 JCL in DL/I DOS/VS 146
 JCL in IMS/VS 144
 restrictions on command 200
 (see also SH20-9173)
 COBOL structures in SO command 200, 240-241
 CODE field on display forms 27
 CODE subject name qualifier 4
 (see also subject code)
 COMMAND action on HEADER form 24
 COMMAND form 34-35
 contents 34
 description 34
 sample form 34
 segmented output with CICS/VS 12 use 34
 commands 179-242
 abbreviations 180-181
 batch and online listed 179
 conventions 181-182
 character set used in 182
 ditto not used in, listed 186
 entering 182
 examples of 187
 general format 183-184
 keywords in 184-186
 summary 179-181
 using ditto in 186-187
 comment, entered with commands 184
 continuation character (+), in commands 182, 184
 COPY command 200-202
 hierarchic sequence used in 200
 (see also SH20-9173)
 purpose 200
 naming and status rules for COPY command 201-202
 copying function, summarized 200
 CRD command 197-198
 (see also CHANGE_RELATIONSHIP_DATA command)

D

D, status reserved for Dictionary data bases 6
 data base-segment relationship keywords 276-286
 DATABASE category defined 1
 keywords in 244-249
 data base type (see DBTP)

DB action on the SYSDEF form 125
 DBAMOD 9
 DBD_IN command 202-203
 restrictions on use 203
 (see also SH20-9173)
 DBD_OUT command 203-205
 use of GSAMDBD 204-205
 DBTP subject name qualifier default 8
 DD command 209-210
 (see also DELETE_DATA command)
 DDUSER category 1
 DDT_OUT command 203-205
 DEFAULTS command 207-208
 example of response to command 208
 default destinations in commands general 184
 in REPORT command 227
 default values, batch forms 147
 defaults, display forms 18
 modifying 23
 defining subjects 1
 definitions in different subject categories 3
 DEL command 209
 (see also DELETE command)
 DELETE action on display forms 24, 29
 on SECONDARY INDEX form 104
 DELETE command 209
 DELETE_DATA command 209-210
 DELETE_RELATIONSHIP command 210-211
 DELETE_RELATIONSHIP_DATA command 212-213
 DELETE_STRUCTURE command 213-214
 caution about use 214
 hierarchic sequence of processing 213
 deleting definitions on display forms 17-18
 DESC action, display forms 30
 DESC field on display forms 29
 DESC in REPORT command 231
 DESC in SCAN command 233
 description defined 2
 in display forms 29
 DESCRIPTION form 35-36
 contents 35
 description 35
 line 001 35
 sample form 36
 use 35-36
 description data record on the Field Definition form 153-154
 on the Field Definition form for PL/I Data 161-162
 on the Segment Definition form 170-171
 DEST defined 184
 in REPORT commands 227
 DETAIL in RPT command 225, 226
 DI command 202-203
 (see DBD_IN command)
 dictionary system messages (see Appendix C)
 DISPLAY action, display forms 22, 24
 display form equivalent reports 227
 display forms ALIAS form 30-31
 CATEGORY-RELATIONSHIPS form 32-33
 COMMAND form 33-35

DESCRIPTION form 35-36
 EXPLANATION form 37
 Extensibility EXPLANATION
 form 38
 EXTENSIBILITY SUBJECT form 39-41
 FIELD form 41-46
 FIELD-SEGMENTS form 46-48
 HEADER forms 20-26
 INQUIRY ATTRIBUTES form 48-50
 JOB form 50-52
 LCHILD form 52-54
 LOG-SEGMENT form 54-56
 LOGICAL DATABASE form 56-58
 LOGICAL DATABASE STRUCTURE
 form 58-61
 MODULE form 61-62
 no-role SEGMENT form 107-111
 PCB form 62-66
 PCB SENSEGS form 66-69
 PCB SENSEG INDEXES form 69-71
 PHY-SEGMENT form 78-81
 PHYSICAL DATABASE form 71-74
 PHYSICAL DATABASE STRUCTURE
 form 74-77
 PIP-SEGMENT form 81-83
 PL/I DATA form 83-85
 PRIMARY INDEX DATABASE form
 85-90
 PROGRAM form 90-91
 PSB form 92-94
 PSB-TRANSACTION form 94-96
 RELATED ENTITIES form 96-98
 RELATIONSHIP DATA form 98-100
 SECONDARY INDEX DATABASE
 form 104-107
 SECONDARY INDEX form 100-104
 Security HEADER form 24-26
 SEGMENT (no-role) form 107-111
 SEGMENT-FIELDS form 112-115
 selected SUBJECT-RELATIONSHIPS
 form 117-119
 SIP-SEGMENT form 115-117
 Standard HEADER form 20-24
 SUBJECT-RELATIONSHIPS
 forms 117-121
 SUBORDINATE/SUPERIOR FIELDS
 form 121-124
 SYSDEF form 124-125
 SYSDEF-DATABASE form 125-127
 SYSDEF-PSB form 127-129
 SYSDEF-TRANSACTION form 129-131
 SYSTEM form 131-133
 TRANSACTION form 133-135
 unselected SUBJECT-RELATIONSHIPS
 form 119-121
 USER DATA form 135-138
 VLC-SEGMENT form 138-139
 XDFLD form 140-142
 display forms, general information
 action area 17
 defaults 18
 explanation text 19
 format 16
 general use 12
 hierarchy of 14-15
 highlighting 19
 kinds of forms 13
 modes of operation 17, 18
 protection of entries 20
 tables 18
 using actions 16
 window forms 19
 DISPLAY mode
 accessing the HEADER form
 in 9, 11

 defined 16
 restrictions on use of 24
 ditto
 defined 186
 examples in commands 187
 use in commands 186
 DL/I DOS/VIS dictionary operating
 instructions 145-147
 DL/I names
 restrictions on use 7
 restrictions in LOGICAL DATABASE
 STRUCTURE form 59
 DL/I segments 106
 DO command 203-205
 (see also DBD_OUT command)
 DOS considerations
 for COBOL_IN 146
 for PLI_IN 146
 DOWN action, display forms 29
 DR command 210-211
 (see also DELETE_RELATIONSHIP command)
 DRD command 212-213
 (see also DELETE_RELATIONSHIP_DATA
 command)
 DTE (see ELEMENT)
 duplicate user names in new systems 4
 duplicate user names controlling 4, 7

E

ECI (Extensibility Control
 Information) 1
 EDIT action, HEADER form 22, 24
 EDIT mode
 accessing the HEADER form
 in 9, 11
 defined 16
 restrictions on use 24
 ELEMENT category
 defined 1
 keywords in 250-259
 element-element relationships
 in AR command 189-190
 keywords for 289
 ellipsis, use of iv
 end record, batch forms
 on the Field Definition form 158
 on the Field Definition form for
 for PL/I Data 166
 on the Segment Definition
 form 173
 on the Text Data form 178
 ENTER key 16
 entering definitions on display
 forms 16
 entries
 in different subject categories 3
 with commands 182
 with display forms 17
 erasing entries, display forms 18
 errors on display forms 13
 EXECUTE command 214
 EXPLAIN action, display forms 29
 EXPLANATION form 37
 contents 37
 description 37
 sample form 37
 use 37
 EXPLANATION form, general
 information 19
 EXPREL action on the
 CATEGORY-RELATIONSHIPS form 33

EXTEND_RELATIONSHIP command 214-216
Extensibility EXPLANATION form 38-39
 contents 38
 description 38
 sample form 38
 use 38-39
EXTENSIBILITY SUBJECT form 39-41
 contents 40
 description 39
 sample form 39
 specialized actions 41
 use 40-41

F

FIELD category (see ELEMENT)
FIELD form 41-46
 contents 41-45
 description 41
 sample form 42
 specialized actions 45-46
 use 45
 use of OCCURS and VALUE
 fields 44
Field Definition form 150-158
 aliases record 154
 basic field attribute record 153
 COBOL attributes record 155-156
 COBOL occurs record 156
 COBOL value record 156
 description 150
 description data record 153-154
 end record 158
 header record 152-153
 PL/I data record 154-155
 sample form 151
 subfield record 157-158
Field Definition form for PL/I
 Data 159-166
 aliases record 162
 basic field attribute record 161
 description 158
 description data record 161-162
 end record 166
 header record 160-161
 PL/I attributes record 163-164
 PL/I data record 162-163
 PL/I picture and initial
 record 164
 sample form 159
 subfield record 165-166
FIELD-SEGMENTS form 46-48
 contents 46-47
 description 46
 sample form 46
 use 47-48
fields on display forms 27-29
FILE category (see DATABASE)
FLDS action, display forms 29
FLUSH command 216
forms, batch (see batch forms)
forms, display (see display forms)
forward and inverse names 184
functions of commands,
 summarized 179-181

G

glossary output, sorted

 under DL/I DOS/VS 147
 under IMS/VS 145
glossary reports 229-231
GSAMDBD in DBD_OUT command 204-205
Guide reports 231-232

H

HEADER forms 20-26
 (see also Security HEADER form
 and Standard HEADER form)
header line on INQUIRY ATTRIBUTES
 form 50
header record
 on the Field Definition
 form 152-153
 on the Field Definition form
 for PL/I Data 160-161
 on the Segment Definition
 form 169-170
 on the Text Data form 174-175
HDR action, display forms 29
hierarchic order
 importance in AR command 189
hierarchic structure
 deleting 213
hierarchy of display forms 14-15
highlighting in display forms 19
hit, in SCAN command 236

I

ID keywords
 defined 186
 in DD command 210
 in DRD command 212-213
 use with FROM and TO in CI and
 CRD commands 195
 relationship data 275-293
 subject data 243-274
IMS/VS considerations
 for PLI_IN 144
 for COBOL_IN 144
indexed field display form 140-142
 (see also XDFLD form)
indexes, sensitive segment, display
 form
 (see PCB SENSEG INDEXES form)
information-only data on display
 forms 9
input text record 177-178
INQATTR action on the FIELD form 45
INQUIRY ATTRIBUTES form 48-50
 contents 49-50
 description 48
 sample form 48
 use 50
INSERT action
 on the LOGICAL DATABASE
 STRUCTURE form 60
 on the PCB SENSEGS form 69
 on the PHYSICAL DATABASE
 STRUCTURE form 77
interactive display forms facility
 (see display forms)
installation-defined categories 1
installation-defined keywords 184-185
installation-defined relationship
 keywords 185

installation-defined relationships,
adding with AR command 193
inverse names 185
IOMOD 9
ISR (indirect subject reference)
report 227-229

J

JCL (job control language)
for dictionary operation
under DL/I DOS/VS 145
under IMS/VS 143
JOB category 1
JOB form 50-51
contents 50
description 50
sample form 51
use 50-51

K

key forms
defined 13
illustrated 14-15
keywords
how to specify 183
id 186
in commands 184-186
in lists 183
installation-defined 184-185
installation-defined
relationship 185
standard 184, 243-274
standard relationship 184, 275-293
sequence attribute 186

L

LANG subject name qualifier default 8
language code (see status code)
language commentary 2
LCHILD action
on the LCHILD form 54
on the PHY-SEGMENT form 81
LCHILD SEGMENT action, display
forms (see LCHILD action)
LCHILD form 52-54
contents 52-53
description 52
sample form 52
specialized actions 54
use 53-54
leading character as
indicator in SCAN command 236
length of entries in
dictionary commands 182
level 66, COBOL 191-192
level 88, COBOL 190
LEVEL in RL command 223
levels, status, suggested use 4
line 001 of description text
on display forms 29
LOG-SEGMENT form 54-56
contents 54-55, 108-109
description 54

sample form 55
specialized actions 56
use 55-56, 109-110

LOGICAL DATABASE form 56-58
contents 56-57
description 56
sample form 57
specialized actions 58
use 57-58
LOGICAL DATABASE STRUCTURE
form 58-61
contents 58
description 58
sample form 59
specialized actions 60-61
use 59-60
lowlight on display forms 19

M

messages in display forms 24
minus sign
used for erasing in display
forms 18
MODE parameter
in the DBD_IN command 203
on batch forms 149
on the LOGICAL DATABASE
STRUCTURE form 58
on the PHYSICAL DATABASE
STRUCTURE form 76
on the SEGMENT-FIELDS
form 113, 114
on the SUPERIOR/SUBORDINATE
FIELDS form 123
modes of operation, display
forms 16
modifying definitions on display
forms 17-18
modifying subject name qualifiers
in commands 237
in general 8
on display forms 23
MODULE category 1
MODULE form 61-62
contents 62
description 61
sample form 61
use 62
MORE-IND action on the PCB SENSEGS
form 69
MOVE action, display forms
on the LOGICAL DATABASE
STRUCTURE form 61
on the PCB SENSEGS form 69
on the PHYSICAL DATABASE
STRUCTURE form 77

N

NAME field on display forms 27
NAME in RPT command 227, 228
names
DL/I 7
PCB 7
naming conventions 7
naming rules, COPY command 201-202
no-role SEGMENT form 107-111

(see also SEGMENT (no-role) form)
notational conventions iv
null value
 in CHANGE_IN command 195
 in CRD command 195

O

OCC (see occurrence number
qualifier)
OCC field on display forms 27
occurrence number qualifier
 defined 4
 modification of, general 7
 modification of, in commands 237
 modification of, on display
 forms 7, 18, 23
 rules for use 6-7, 28
 use in controlling duplicate
 user names 4
 valid values 6, 28
OCR subject name qualifier
 default 8
online modes 9
online operation of dictionary
 under CICS/VS 10-12
 under IMS/VS 9-10
operating modes
 display forms 16
output of batch operations 147
output of online operations
 under CICS/VS 12
 under IMS/VS 10
output of REPORT command 224
output of SCAN command 235-236
output, routing from online
operations
 under CICS/VS 12
 under IMS/VS 10

P

PA1 key, use on the COMMAND form 35
PARENT in RL command 223
parentheses around list of keyword
values 183
PCB category
 defined 1
 keywords in 260-262
PCB form 62-66
 contents 62-65
 description 62
 sample form 63
 use 65-66
PCB names
 meaning of occurrence
 qualifier 4, 7
 restrictions on 7
PCB-segment relationships
 in the AR command 192
 relationship keywords
 286-287
PCB SENSEGS form 66-69
 contents 66-67
 description 66
 sample form 66
 specialized actions 69
 use 68-69
PCB SENSEG INDEXES form 69-71

 contents 70
 description 69
 sample form 70
 use 70-71
PCB sensitive segments
 (see PCB SENSEGS form)
PHY-SEGMENT form 78-81
 contents 78-80, 108-109
 description 78
 sample form 78
 specialized actions 81
 use 80-81, 109-110
PHYSICAL DATABASE form 71-74
 contents 71-74
 description 71
 sample form 71
 specialized actions 74
 use 74
PHYSICAL DATABASE STRUCTURE
form 74-77
 contents 75-76
 description 74
 sample form 75
 specialized actions 77
 use 76-77
physical segment display form
 (see PHY-SEGMENT form)
PI command 219
 (see also PSB_IN command)
PIP-SEGMENT form 81-83
 contents 82, 108-109
 description 81
 sample form 82
 specialized actions 83
 use 83, 109-110
PL/I action on SEGMENT forms 111
PL/I attributes record 161-162
PL/I data 2
PL/I DATA form 83-85
 contents 84-85
 description 83
 sample form 84
 use 85
PL/I Data record
 on the Field Definition
 form 154-155
 on the Field Definition form for
 PL/I Data 162-163
 on the Segment Definition
 form 171-172
PL/I output with SO command 240
PL/I picture and initial record 164
PL/I structures with
STRUCTURES_OUT 218, 240-241
PLI_IN command 217-218
 JCL for DL/I DOS/VS 146
 JCL for IMS/VS 144
 restrictions 218
 (see also SH20-9173)
 use of LANGUAGE operand 218
plus sign as continuation
character 182, 184
PO command 219-220
 (see also PSB_OUT command)
 primary index pointer display
 form
 (see PIP-SEGMENT form)
POS BEFORE SEGMENT fields
 on the LOGICAL DATABASE
 STRUCTURE form 61
 on the PCB SENSEGS form 69
 on the PHYSICAL DATABASE
 STRUCTURE form 77
PRIMARY INDEX DATABASE form 85-90
 contents 85-88

- description 85
- sample form 86
- use 88-90
- primary index pointer display
 - form 81-83
 - (see also PIP-SEGMENT form)
- primary name 2
- PROC action on display forms 18, 29
- processing control functions
 - summarized 179
- PROGRAM category
 - defined 1
 - keywords in 169
- production status, use of 4
- PROGRAM form 90-91
 - contents 90
 - description 90
 - sample form 91
 - use 91
- programming language data structure
 - processing summarized 181
- PSB action on the SYSDEF form 125
- PSB category
 - defined 1
 - keywords in 270-271
- PSB form 92-94
 - contents 92-94
 - description 92
 - sample form 92
 - specialized actions 94
 - use 94
- PSB-TRANSACTION form 94-96
 - contents 94
 - description 94
 - sample form 95
 - use 95-96
- PSB-transaction relationship
 - keywords 293
- PSB_IN command 219
 - restrictions 219
 - (see also SH20-9173)
- PSB_OUT command 219-220
- PUNCH command 220-221

Q

- qualifiers, subject name
 - (see subject name qualifiers)
- QUIT action, display forms 23, 24

R

- RECALCULATE_SEGMENT command 221-222
- RECORD category (see SEGMENT)
- REGEN action on display forms 30
- RELATED ENTITIES form 96-98
 - contents 97
 - description 96
 - sample form 96
 - use 97-98
- RELATED FIELDS action on the FIELD
 - form 45-46
- related fields display form
 - (see SUBORDINATE/SUPERIOR FIELDS form)
- relationship, alias (see alias)
- relationship data 2
 - in display forms 14-15
 - keyword values 275-294

- not all deleted by DRD
 - command 212-213
 - subject categories for which entered 3
- RELATIONSHIP DATA form 98-100
 - contents 99
 - description 98
 - sample form 98
 - specialized actions 100
 - use 99-100
- relationships
 - defined 2
 - keywords 275-293
- RELATIONSHIPS action on the EXTENSIBILITY SUBJECT form 41
- relationships established by PSB_IN 219
- relationships possible between categories 275-293
- RELDATA action
 - on the selected SUBJECT-RELATIONSHIPS form 119
 - on the unselected SUBJECT-RELATIONSHIPS form 121
- RELOCATE command 222-224
 - restrictions 224
 - use of LEVEL 223
 - use of PARENT 223
- REPORT command 224-232
 - display form equivalent reports 227
 - glossary reports 229-231
 - Guide reports 231-232
 - indirect subject reference reports 227-229
 - subject-specific reports 224-227
- restrictions
 - on the COBOL_IN command 200
 - on the DBD_IN command 203
 - on the PLI_IN command 218
 - on the PSB_IN command 219
 - on the RELOCATE command 223-224
 - on the status of related subjects in the AR command 188-189
 - on user names 4, 7
- REUSE action
 - general use 30
 - on SEGMENT forms 109
- RL command 222-224
 - (see also RELOCATE command)
- role in SEGMENT forms 107, 108, 109
- routing of output from online operations
 - under CICS/VS 12
 - under IMS/VS 10
- RPT command 224-232
 - (see also REPORT command)
- RPTNAME in RPT command 227, 228
- RS command 221-222
 - (see also RECALCULATE_SEGMENT command)
- RTN action, display forms 30

S

- SC command 232-236
 - (see also SCAN command)
- SCAN and REPORT functions
 - summarized 181
- SCAN command 232-235
 - character strings in command 236
- SEC INDEX NAME action
 - on the SECONDARY INDEX DATABASE

- form 107
 - on the SECONDARY INDEX form 104
- SECONDARY INDEX DATABASE
 - form 104-107
 - contents 105-107
 - description 104
 - sample form 105
 - specialized actions 107
 - use 107
- SECONDARY INDEX form 100-104
 - contents 101-103
 - description 100
 - sample form 101
 - specialized actions 104
 - use 103-104
- secondary index pointer display form
 - (see SIP-SEGMENT form)
- secondary name (see alias)
- security
 - general information 8
 - HEADER form 24-26
 - in display forms 24-25
 - messages 24
- Security HEADER form 24-26
 - contents 24
 - description 24
 - sample form 25-26
 - security messages 24
 - use 24
- SEGMENT (no-role) form 106-110
 - contents 107-108
 - description 106
 - sample form 109
 - specialized actions 110
 - use 108, 110
- SEGMENT category
 - defined 1
 - keywords in 263-268
- Segment Definition form 167-173
 - aliases record 171
 - basic segment attribute
 - record 170
 - description 167
 - description data record 170-171
 - end record 173
 - header record 169-170
 - PL/I data record 171-172
 - related field record 172-173
 - sample form 168
- segment-element relationships
 - in AR command 191
 - keywords for 288
- SEGMENT-FIELDS form 112-115
 - contents 112-114
 - description 112
 - sample form 113
 - use 114-115
- segment-PCB relationships
 - (see PCB-segment relationships)
- segments, sensitive, display form
 - (see PCB SENSEGS form)
- SEGS action on the FIELD form 46
 - selected SUBJECT-RELATIONSHIPS
 - form 116-119
 - (see also SUBJECT-RELATIONSHIPS form (selected))
- SENSEG indexes display form
 - (see PCB SENSEG INDEXES form)
- sensitive segment indexes display form
 - (see PCB SENSEG INDEXES form)
- sensitive segments, PCB
 - (see PCB SENSEGS form)
- sequence of display forms 14-15
- SET commands 237-238
- shutting down the Dictionary under
 - CICS/VS 11-12
 - SIGN_ON command 234-235
 - signing on and off display
 - forms 9-12
 - SIP-SEGMENT form 115-117
 - contents 115, 108-109
 - description 115
 - sample form 116
 - specialized actions 117
 - use 115-116, 109-110
 - SO command 240-242
 - (see also STRUCTURES_OUT command)
 - sorted glossary output
 - under DL/I DOS/VS 147
 - under IMS/VS 145
 - STAGE_1_OUT command 239-240
 - Standard HEADER form 19-23
 - contents 20-21
 - description 19
 - sample form 20
 - specialized actions 23
 - use 21-23
 - standard categories 2
 - standard relationships 2
 - standard relationships in the
 - AR command 184
 - starting an online session
 - under CICS/VS 10-11
 - under IMS/VS 9
 - STAT (see status subject name
 - qualifier)
 - STAT subject name qualifier
 - default 8
 - status rules in COPY command 201
 - status subject name qualifier
 - D status reserved 4
 - defined 4
 - in Dictionary structures 4
 - modification of, general 8
 - modification of, in commands 237
 - modification of, on display
 - forms 8, 18, 24
 - P, for production 4
 - rules for use 6-7, 28
 - valid values 6, 28
 - STRUC action
 - on the LOGICAL DATABASE form 57
 - on the PHYSICAL DATABASE form 74
 - STRUCTURES_OUT command 240-242
 - subfield record
 - on the Field Definition
 - form 157-158
 - on the Field Definition form
 - for PL/I Data 165-166
 - on the Segment Definition
 - form 172-173
 - on the Text Data form 177-178
 - subject categories 1
 - definitions in 1
 - ECI 1
 - installation-defined 1
 - standard 1
 - subject code
 - defined 4
 - modification of, general 8
 - modification of, in commands 237
 - modification of, on display
 - forms 8, 18, 24
 - rules for use 6-7, 28
 - valid values 6, 28
 - subject data 2
 - subject name
 - defined 4
 - format 4
 - in display forms 27

- rules in commands 6-7
- rules in display forms 28
- subject name qualifiers
 - defaults 8
 - defined 3-4
 - on display forms 18, 23
- SUBJECT-RELATIONSHIPS form
 - (selected) 117-119
 - contents 118
 - description 117
 - sample form 117
 - specialized actions 119
 - use 118-119
- SUBJECT-RELATIONSHIPS form
 - (unselected) 119-121
 - contents 119-120
 - description 119
 - sample form 120
 - specialized actions 121
 - use 120-121
- subject-specific reports 224-227
- SUBJECT1 action on the RELATIONSHIP DATA form 100
- SUBJECT2 action
 - on the RELATIONSHIP DATA form 100
 - on the selected SUBJECT-RELATIONSHIPS form 119
 - on the unselected SUBJECT-RELATIONSHIPS form 121
- subordinate display forms,
 - common 14-15
- subordinate elements in AR command 190
- SUBORDINATE/SUPERIOR FIELDS form 121-124
 - contents 122-123
 - description 121
 - sample form 122
 - use 123-124
- superior elements in AR command 190
- SYSDEF category 1
- SYSDEF form 124-125
 - contents 125
 - description 124
 - sample form 124
 - specialized actions 125
 - use 125
- SYSDEF-DATABASE form 125-127
 - contents 126
 - description 125
 - sample form 126
 - use 126-127
- SYSDEF-database relationship keywords 290
- SYSDEF-PSB form 127-129
 - contents 127-128
 - description 127
 - sample form 128
 - use 128-129
- SYSDEF-PSB relationship keywords 291
- SYSDEF-TRANSACTION form 129-131
 - contents 129-131
 - description 129
 - sample form 130
 - use 131
- SYSDEF-transaction relationship keywords 292-293
- SYSTEM category 1
- system categories
 - defined 3
 - keywords in 269-274
- system definition category 1
 - (see also SYSDEF)

- system description functions summarized 180
- system-definition display form
 - (see SYSDEF form)
- system-definition-data base display form
 - (see SYSDEF-DATABASE form)
- system-definition-PSB display form
 - (see SYSDEF-PSB form)
- system-definition-transaction display form
 - (see SYSDEF-TRANSACTION form)
- SYSTEM form 131-133
 - contents 132
 - description 131
 - sample form 132
 - use 132-133

T

- tables in display forms 18
- terminating an online session
 - under CICS/VS 11
 - under IMS/VS 10
- test status, use of 4
- Text Data form 174-178
 - end record 178
 - header record 176-177
 - input text record 177-177
- TOP action, display forms 30
- TRAN action
 - on the PSB form 94
 - on the SYSDEF form 125
- TRANSACTION category
 - defined 1
 - keywords in 272-273
- TRANSACTION form 133-135
 - contents 133-135
 - description 133
 - sample form 134
 - use 135

U

- universal match character
 - in SCAN command 236
- unselected SUBJECT-RELATIONSHIPS form 118-120
 - (see also SUBJECT-RELATIONSHIPS form (unselected))
- update and COPY functions summarized 180-181
- User Data 2
- USER DATA form 135-138
 - contents 136
 - description 135
 - sample form 136, 137
 - use 136-138
 - restricted to range 1-255
- user names
 - defined 4
 - mixed with subject names in SO command 241
 - restricted to 8 characters 6
 - restrictions in programming languages 4, 7
 - restrictions on display forms 28

rules in commands 4, 6-7
rules in display forms 28
valid characters 4
USERDATA-NO action, display
forms 30
using FROM and TO in CI
and CRD commands 195-196

V

virtual logical child display
form
(see VLC-SEGMENT form)
VLC-SEGMENT form 138-139
contents 139, 108-109
description 138
sample form 138
specialized actions 139
use 139, 109-110

W

window forms, display forms 19

X

XDFLD action
on the SECONDARY INDEX form 104
on the SIP-SEGMENT form 117
XDFLD form 140-142
contents 140-141
description 140
sample form 140
use 141-142
XR command 214-216
(see also EXTEND_RELATIONSHIP command)

IBM Data Processing Division, International Sales and Service, White Plains, N.Y., U.S.A. 10604



International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Please use pressure sensitive or other gummed tape to seal this form.

List TNLs here:

If you have applied any technical newsletters (TNLs) to this book, please list them here:

Last TNL _____

Previous TNL _____

Previous TNL _____

Fold on two lines, tape, and mail. No postage necessary if mailed in the U.S.A. (Elsewhere, any IBM representative will be happy to forward your comments.) Thank you for your cooperation.

Reader's Comment Form

Fold and tape

Please do not staple

Fold and tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
P.O. Box 50020
Programming Publishing
San Jose, California 95150



Fold and tape

Please do not staple

Fold and tape



International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601