

IBM File Export for z/OS



Reference Manual

Version 1 Release 1



IBM File Export for z/OS



Reference Manual

Version 1 Release 1

Note: Before using this information and the product it supports, be sure to read the general information under Notices.

Third Edition (January 2005)

This edition applies to Version 1 Release 1 of IBM File Export for z/OS (product number 5697-I12) and to all subsequent releases and modifications until otherwise indicated in new editions.

©Copyright International Business Machines Corp.2003, 2005. ©Copyright Rocket Software, Inc.2003-2005. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	vii
---------------	-----

About this book	ix
Who should read this book	ix
Conventions used in this book	ix
Terminology used in this book	x
Service updates and support information.....	x
Using LookAt to obtain message explanations.....	x
Where to find information	x
Accessibility features	x
How to send your comments	xi

Part 1. Introduction

1

Chapter 1. Introduction	3
Overview	3
IBM File Export Interfaces	4
Sources, targets, and associated ddnames.....	4
Data manipulation	4
The File Export repository	5
Defining relationships in IBM File Export.....	5
IBM File Export information	6

Part 2. Installation and Customization 7

Chapter 2. Installing IBM File Export for z/OS.....	9
Hardware requirements	9
Software requirements	9
Installing IBM File Export for z/OS	9
Verifying that your installation is complete	10
Security	10

Chapter 3. Customization Tasks.....	11
Customizing the IBM File Export CLISTs	11
CLIST Member ABX	11
CLIST member ABXCLIST	12
Customizing the Global Options Block	13
Security	13
Changing Global Options.....	13
Modifying CSECT ABXOGBL0	13
Customizing DB2 for use with IBM File Export.....	14
Security	14
Creating a DB2 Options Block	14
Customizing DB2 BIND jobs	16
Job ABXBIND	16
Job ABXBIND3	17
Customizing IMS for use with IBM File Export	20
Security	20
Creating an IMS Options Block	21
Modifying CSECT ABXOIMS0	21
Customizing IBM File Export to use auditing	22
Modifying an ISPF panel to start IBM File Export	23

Part 3. Start-up and Configuration .. 25

Chapter 4. Start-up and Configuration using the IBM File Export User Interface.....	27
The IBM File Export Main Menu.....	27
Administering the IBM File Export repository	28
Selecting and managing your IBM File Export sessions	28
Configuring IBM File Export global options	29

Part 4. The File Export Repository .. 33

Chapter 5. Working with the IBM File Export Repository	35
Summary.....	35
The Repository Management panel	35
Creating a new repository	36
Deleting an existing repository	37
Opening an existing repository	37
Create a storage structure	38
Delete a storage structure	39
Refresh the storage structure list	39
Define record layouts.....	39
Open a storage structure.....	39
Add or edit a storage structure	39
Working with a DB2 storage structure	45
Open a DB2 Storage Structure	45
Working with an IMS storage structure	47
Open an IMS Storage Structure	47
Working with VSAM or Sequential records	49
Working with record layouts.....	49
Working with DB2 record layouts	52
Working with IMS record layouts	63
Working with VSAM and sequential record layouts	78
Managing Record Relationships	92
Specifying DB2 table and RI definitions.....	94
Using the parser.....	96

Part 5. Working with IBM File Export 99

Chapter 6. Getting Started with the IBM File Export ISPF Interface	101
Exporting data using IBM File Export for z/OS.....	101
What sources and targets can I use?	101
Using the Session Management (ABX\$SMNG) panel	102
Specifying selection criteria	103

Chapter 7. Exporting DB2 Data	105
Summary.....	105
Specify DB2 product options	105
Select a DB2 session.....	106
Specify the subsystems and LOADLIBs for your source and target	107
Define Relationships	109
Specify and manipulate query data	113
Edit source tables	115
Edit source columns.....	116
Edit selection criteria	118

Modify target columns	119
Edit query relationships	122
Edit Query Criteria	123
Specify target table information	124
Specify intermediate data set DD statements	126
Generate JCL	128

Chapter 8. Exporting IMS data 133

Select the IMS source and target types.	133
Summary	134
The IMS Menu	135
Specify IMS session settings	136
Specify source database	140
If your source processing option is IMS or DLI	141
If your source processing option is DEDB...	142
If your source processing option is DBB or BMP..	143
Specify target database	144
If your target processing option is IMS or DLL....	144
If your target processing option is DEDB, DBB, or	
BMP.....	146
If your target processing option is FEUNLOAD..	147
Define variables and literals	148
Define relationships	150
Define segment specification and manipulation	154
Create DD statements	166
Generate JCL	170

Chapter 9. Exporting Sequential Data or VSAM Data 173

Summary	173
The Sequential and VSAM Menu	174
Specify session settings.	175
Specify source and target data sets	178
Define variables and literals	181
Define relationships	183
Define field specification and manipulation	187
Define type and select statement specification and	
manipulation	191
Generate JCL	198

Part 6. Using Control Statements . . 201

Chapter 10. Creating JCL Using Control Statements 203

Control statement syntax	203
General order of control statements	203
IMS example	203
Names used in control statements.	204
Applicability of control statements based on type of	
source data	204
Control statement syntax, definitions, and	
descriptions	206
AGN	207
ALTID	207
ALTRRC00.	207
AUDIT	207
BKO	208
BUF	208
COLLECT	208
COLUMN	209
CPUTIME	210
DATE	210
DBDLIB	212
DBRC	212

DFSVSAMP	213
DIRCA	213
DLTDUPS	213
DOPTLIB	214
DPSBMAC	214
DPSBPFX	214
DPSBSFX	215
DYNPSB	215
EXCLUDE	215
EXCPVR	216
EXITACTIVE	216
EXITNAME	216
FIELD	217
FMTO	219
GSGNAME	220
IMSID	220
IMSPLEX	220
INPCB#	221
INSERT	221
IRLM	221
IRLMNAME	222
LAST_SEGMENT	222
LIMIT	222
LITERAL	222
LOCKMAX	224
MON	224
NBA	224
OBA	225
OFIELD	225
OPT	228
OPTIONS	228
OUTPCB#	232
PARDLI	232
PROCOPT	232
PSBNAME	233
RELATIONSHIP	233
RESLIB1	236
RESLIB2	236
RESLIB3	236
RESULTSET	236
SEGMENT	237
SELECT statements	239
SKIP	244
SOURCEDB2	244
SOURCEDBD	244
SOURCEDD	245
SSM	245
STIMER	245
SWAP	246
TABLE	246
TARGETDB2	248
TARGETDBD	248
THEN INSERT	248
TYPE	251
VARIABLE	254
WUNIT	256

Chapter 11. Additional Information about Control Statement Syntax 257

Including comments in your JCL	257
About scrambling	257
Specification of IBM File Export data types.	257
Understanding scope	257
Scope of fields, literals, and variables	258
Scope of SKIP and LIMIT	258
Limiting output record counts using SKIP and	

LIMIT	258
Global specification of SKIP and LIMIT	258
Non-global specification of SKIP and LIMIT	259
Inserting data into IMS databases using the File Export	
UNLOAD file	259
About ABXFELD0	259
Requirements and restrictions	259
JCL requirements	260
ABXFELD0 control statements	260
Working with variable length records	261
Variable length input records and the use of the	
asterisk (*)	261
Using slack bytes when creating output files	261
Understanding INSERT and THEN INSERT	262
Understanding Occurs Depending On	262
Specifying MINIMUM, MAXIMUM, PRINT, or	
SPUFI	262

Chapter 12. Formatting Output Records	263
Output record formatting	263
How use of the Cobol parser affects output alignment	264

Part 7. Appendixes. 265

Appendix A. DDnames used by IBM File	
Export for z/OS.	267
External data sets required when working with IMS Fast	
Path data	270

Appendix B. Installation Verification	
Programs	273
IVP for DB2	273
IVP for IMS	273
IVP for VSAM	274
IVP for sequential files	274

Appendix C. IBM File Export Global	
Options	277
Parameters	277

Appendix D. IBM File Export DB2	
Options	281
Parameters	281

Appendix E. IBM File Export IMS	
Options	283
Parameters	283
Understanding static and dynamic PSBs	290
Using static PSBs	291
Using dynamic PSBs	291

Appendix F. Writing an I/O Exit.	293
Exit Protocol	293
Writing your exit	294
Programming your I/O exit	294
Using the I/O exit control block	297
Function codes and the flow of I/O exit processing ..	298
Information that can be returned from the exit	298
Exit functions	298
The Initialization call	298
The Termination call	299
The Open call	299
The Close call	299
Read and Write calls	299

Appendix G. Troubleshooting.	301
LOBs	301
Output record formatting when using the COBOL	
parser	301

Appendix H. Notices	303
Trademarks	305

Glossary	307
-----------------------	------------

Index	309
--------------------	------------

Figures

1. IBM File Export Main Menu (ABX\$MAIN).	27	66. Source Tables (ABX\$DBTL) panel.	115
2. IBM File Export Repository Management (ABX\$RMNG) panel.	28	67. Query Columns (ABX\$DBCL) panel	116
3. IBM File Export Session Management (ABX\$SMNG) panel	29	68. Query Columns (ABX\$DBCL) panel with pop-up	117
4. Settings (ABX\$GOPT) Menu.	30	69. Columns Selection (ABX\$DBCS) panel	118
5. Repository Management (ABX\$RMNG) panel	35	70. Selection Criteria (ABX\$CLST) Panel	119
6. Create Repository (ABX\$ZCRR) panel	36	71. Target Columns (ABX\$DBTC) panel	120
7. Delete Repository (ABX\$ZDLR) panel.	37	72. Edit Target Columns (ABX\$DTC1 or ABX\$DTC2) panel	121
8. Repository Selection (ABX\$ZRNM) panel	37	73. Edit Relation (ABX\$RRED) panel	122
9. Storage List (ABX\$ZSTL) panel.	38	74. Statement List (ABX\$RRLS) panel	123
10. Create Storage (ABX\$ZSTR) panel	40	75. Edit Query Criteria (ABX\$QCR) panel.	124
11. Edit DB2 Storage (ABX\$ZSE1) panel	41	76. Target Specification (ABX\$DBQT) panel.	125
12. Edit IMS Storage (ABX\$ZSE2) panel	42	77. DD Statement Definition (ABX\$DDE1) panel	127
13. Edit IMS Storage (ABX\$ZSE5) panel	43	78. Data Set Allocation Parameters (ABX\$DDWZ) panel	128
14. Edit VSAM or Sequential Storage (ABX\$ZSE3) panel	44	79. JCL Generation (ABX\$DBJC) panel	129
15. Edit VSAM or Sequential Storage (ABX\$ZSE3) panel	45	80. Member Selection List (ABX\$MEML) panel	130
16. Table List (ABX\$ZTLS) panel	46	81. ABX\$DSSL panel.	130
17. Tables Selection (ABX\$ZTBS) panel.	47	82. IMS Source and Target Types (ABX\$IMST) panel	134
18. Segment List (ABX\$ZSLS) panel.	48	83. IMS Menu (ABX\$IMSW) panel	135
19. Segment Selection (ABX\$ZSGS) panel.	49	84. IMS Session Settings (ABX\$IOPT) panel	137
20. Record Layouts (ABX\$ZRDL) panel	50	85. IMS Source Database Selection (ABX\$FFDB) panel	141
21. Edit Record (ABX\$ZSE4) panel	51	86. ABX\$DSLS panel	141
22. Create Record (ABX\$ZREC) panel	51	87. IMS Source Database Specification (ABX\$FPSR) panel	142
23. Column List (ABX\$ZCNL) panel	52	88. ABX\$DSLS panel	142
24. Field List (ABX\$ZFDL) panel	53	89. IMS Source Database Specification (ABX\$FPTG) panel	143
25. Edit Simple Field (ABX\$ZESF) panel	54	90. ABX\$DSLS panel	144
26. Edit Group Field (ABX\$ZEGF) panel.	56	91. IMS Target Database Selection (ABX\$FFDB) panel	145
27. Create Simple Field (ABX\$ZCSF) panel.	58	92. ABX\$DSLS panel	145
28. Create Group Field (ABX\$ZCGF) panel	60	93. IMS Target Database Specification (ABX\$FPTG) panel	146
29. Table List (ABX\$ZTLS) panel	62	94. ABX\$DSLS panel	147
30. Record Layouts (ABX\$ZTRL) panel.	62	95. IMS Target Database Specification (ABX\$FPTG) panel	148
31. Field List (ABX\$ZFDD) panel	64	96. Variables and Literals (ABX\$VLST) panel	149
32. Field List (ABX\$ZFDL) panel	65	97. New Literal (ABX\$VARS) panel	149
33. Edit Simple Field (ABX\$ZESF) panel	66	98. Relationship Specification (ABX\$RELS) panel	150
34. Edit Group Field (ABX\$ZEGF) panel.	68	99. Filter Specification (ABX\$RELF) panel.	151
35. Create Simple Field (ABX\$ZCSF) panel.	70	100. Relation Content (ABX\$RELC) panel.	152
36. Create Group Field (ABX\$ZCGF) panel	72	101. Segments (ABX\$SEGM) panel	154
37. Segment List (ABX\$ZSLS) panel.	74	102. Segment Fields (ABX\$FSEL) panel	155
38. Record Layouts (ABX\$ZFRL) panel.	74	103. Field Specification (ABX\$NFLD) Panel	157
39. Manage Record Layouts (ABX\$ZMRL) panel	75	104. Record Type List (ABX\$SQLT) Panel.	159
40. Selection Criteria (ABX\$ZSCR) panel	76	105. Record Type (ABX\$SQLT) Panel.	160
41. Record Layouts Selection (ABX\$ZRDS) panel	77	106. Select Statement List (ABX\$SELS) Panel	161
42. Field List (ABX\$ZFDD) panel	78	107. Edit Select Statement (ABX\$SELE) Panel.	162
43. Field List (ABX\$ZFDL) panel	79	108. Selected Fields (ABX\$SLST) Panel	163
44. Edit Simple Field (ABX\$ZESF) panel.	81	109. New Field Selection (ABX\$FLST) Panel.	164
45. Edit Group Field (ABX\$ZEGF) panel.	82	110. Selection Criteria (ABX\$CLST) Panel	164
46. Create Simple Field (ABX\$ZCSF) panel.	84	111. Edit Relation (ABX\$RRED) Panel	165
47. Create Group Field (ABX\$ZCGF) panel	86	112. DD Statements List (ABX\$IMSD) panel	166
48. Record Layouts (ABX\$ZFRL) panel.	88	113. DD Statement Definition (ABX\$DDE1) panel.	167
49. Manage Record Layouts (ABX\$ZMRL) panel	89	114. Data Set Allocation (ABX\$DDWZ) panel	168
50. Selection Criteria (ABX\$ZSCR) panel	90	115. Sort control statements (ABX\$DXS) panel	169
51. Record Layouts Selection (ABX\$ZRDS) panel	91	116. JCL Generation for IMS Session (ABX\$IMSJ) panel	170
52. Record Relations (ABX\$ZREL) panel.	92	117. ABX\$DSSL panel.	171
53. Edit Relation (ABX\$ZERL) panel	93	118. Sequential & VSAM Menu (ABX\$SEQW) panel	174
54. Relation Field List (ABX\$ZFRE) panel.	94	119. Session Settings (ABX\$FOPT) panel.	176
55. Import DB2 Definitions (ABX\$DBRI) panel	95	120. DD Statement Definition (ABX\$DDE1) panel	177
56. Parser Configuration (ABX\$ZPRS) panel	97	121. Data Set Allocation Parameters (ABX\$DDWZ) panel	178
57. IBM File Export Session Management (ABX\$SMNG) panel	102	122. Sequential Data Set Specification (ABX\$VSD) panel	179
58. DB2 Options (ABX\$DOPT) panel.	106		
59. DB2 Menu (ABX\$DB2W) panel.	107		
60. DB2 Source and Target (ABX\$DBST) panel	108		
61. Relationship Specification (ABX\$RELS) panel	109		
62. Filter Specification (ABX\$RELF) panel	110		
63. Relation Content (ABX\$RELC) panel.	111		
64. Queries (ABX\$DBQL) panel	113		
65. Edit Query (ABX\$DBQE) panel	114		

123. DD Statement Definition (ABX\$DDE4) panel ..	180	133. Record Type (ABX\$SQLT) panel.	193
124. DD Statement Definition (ABX\$DDE2) panel. ..	181	134. Select Statement List (ABX\$SELS) Panel	194
125. Variables and Literals (ABX\$VLST) panel.	182	135. Edit Select Statement (ABX\$SELE) Panel.	195
126. New Variable (or Literal) (ABX\$VARS) panel..	183	136. Selected Fields (ABX\$SLST) Panel	196
127. Relationship Specification (ABX\$RELS) panel..	183	137. New Field Selection (ABX\$FLST) Panel.	197
128. Filter Specification (ABX\$RELF) panel	185	138. Selection Criteria (ABX\$CLST) Panel	197
129. Relation Content (ABX\$RELC) panel.	186	139. JCL Generation (ABX\$JCLG) panel	198
130. Fields (ABX\$FSEQ) panel	188	140. Member Selection List (ABX\$MEML) panel	199
131. Field Specification (ABX\$NFLD) Panel	189	141. ABX\$DSSL panel	200
132. Record Type List (ABX\$SQTL) Panel.	192		

About this book

This book provides instructions for installing IBM File Export for z/OS. It is also a reference that you can use to review and understand the product's user interface and its batch facility.

This book is designed to help database administrators, system programmers, application programmers, and system operators perform these tasks:

- Plan for the installation of IBM File Export for z/OS
- Install and operate IBM File Export for z/OS
- Customize your IBM File Export for z/OS environment
- Diagnose and recover from IBM File Export for z/OS problems
- Use IBM File Export for z/OS with Sequential, VSAM, DB2, or IMS data

Always check the DB2 and IMS Tools Library page for the most current version of this publication: www.ibm.com/software/data/db2imstools/library.html

Who should read this book

This book is intended for those persons responsible for installing, customizing, and using IBM File Export for z/OS and assumes a working knowledge of:

- The OS/390[®] or z/OS[®] operating system
- ISPF
- SMP/E
- IMS
- DB2

Conventions used in this book

This book uses the following highlighting conventions:

- **Boldface type** indicates commands or user interface controls such as names of fields, folder, icons, or menu choices.
- **Monospace type** indicates examples of text that you enter exactly as shown.
- *Italic type* indicates variables that you should replace with a value. It is also used to indicate book titles and to emphasize significant words.

The following labels identify significant elements within this book:

- **Definition:** is used to identify and define terminology unique to this product.
- **Example:** is used to identify example code or scenarios.
- **In this chapter:** is used to identify the significant subsections within each chapter.
- **Recommendation:** is used to provide guidance when more than one option is available.
- **Related reading:** is used to refer you to other publications that contain relevant information.
- **Restriction:** is used to identify a restriction or limitation with this product or an associated procedure.

Terminology used in this book

In this book, IBM File Export for z/OS is referred to as “IBM IBM File Export.” In cases where the context makes the meaning clear, IBM IBM File Export is referred to as “IBM File Export.”

The following terms are used as indicated:

DB2® Represents either the DB2 licensed program or a particular DB2 subsystem.

IMS Represents the IMS licensed program.

In general, for simplicity, the term *file* as used in this manual refers to IMS, VSAM or sequential files, as well as DB2 tables, unless these are specifically differentiated.

Service updates and support information

To find service updates and support information, including software Fix Packs, PTFs, Frequently Asked Question (FAQs), technical notes, troubleshooting information, and downloads, refer to the following Web page:

<http://www.ibm.com/software/awdtools/fileexport/>

Using LookAt to obtain message explanations

LookAt is an online facility that displays explanations for most messages that you encounter, as well as for some system abends and codes. You can access LookAt:

- From the Internet at:
www.ibm.com/eserver/zseries/zos/bkserv/lookat/
- From anywhere in z/OS where you can access a TSO/E command line (for example, a TSO/E prompt, ISPF, or z/OS UNIX System Services running OMVS). To use LookAt as a TSO/E command, LookAt must be installed on your host system. Obtain the LookAt code for TSO/E from a disk on your z/OS Collection (SK3T-4269) or from the LookAt ftp site:
[ftp.software.ibm.com/ps/products/ibmreader/tools/lookat/ZOS](ftp://software.ibm.com/ps/products/ibmreader/tools/lookat/ZOS)
- From your Palm™ VIIx personal data assistant (PDA). To use LookAt from your Palm VIIx PDA, LookAt must be installed on the PDA. Obtain the LookAt code from a disk on your z/OS Collection (SK3T-4269) or from the LookAt ftp site:
[ftp.software.ibm.com/ps/products/ibmreader/tools/lookat/PALM/](ftp://software.ibm.com/ps/products/ibmreader/tools/lookat/PALM/)

Where to find information

The Web page listed below provides the most current product documentation that you can view, print, and download.

<http://www.ibm.com/software/awdtools/fileexport/>

Accessibility features

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use a software product successfully. The major accessibility features in IBM IBM File Export enable users to:

- Use assistive technologies such as screen readers and screen magnifier software. Consult the assistive technology documentation for specific information when using it to access z/OS interfaces.
- Customize display attributes such as color, contrast, and font size.
- Operate specific or equivalent features using only the keyboard. Refer to the z/OS ISPF User's Guide for information about accessing ISPF interfaces. This guide describes how to use ISPF, including the use of keyboard shortcuts or function keys (PF keys), includes the default settings for the PF keys, and explains how to modify their functions.

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other IBM File Export documentation, print and fill out the reader comment form located at the back of this book. You can use the following methods to return the form:

- Give it to your local IBM branch office or IBM representative.
- Send it to the address printed on the reader comment form.
- Send your comments by e-mail to comments@us.ibm.com.

Part 1. Introduction

Chapter 1. Introduction	3	The File Export repository.....	5
Overview	3	Defining relationships in IBM File Export	5
IBM File Export Interfaces	4	IBM File Export information.....	6
Sources, targets, and associated ddnames.....	4		
Data manipulation	4		

This part contains an overview of IBM File Export for z/OS.

Chapter 1. Introduction

IBM File Export for z/OS is a high performance data gathering, manipulation, and file- or table-generation solution used to export related sets of existing data from similar or dissimilar sources. Use IBM File Export to extract DB2, IMS, VSAM, and sequential data from one or more sources and create one or more targets. (In general, for simplicity, the term file as used in this manual refers to IMS, VSAM or sequential files, as well as DB2 tables unless these are specifically differentiated.)

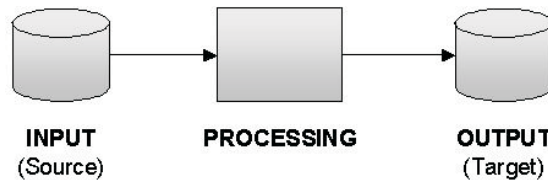
IBM File Export for z/OS design objectives:

- Simplify application development
- Expedite development testing
- Improve productivity by saving time

Overview

Application developers typically know and understand their data and that data's relationships. However, integrating data from different sources can be a problem. The term *source* refers to input data, such as an existing DB2 table or an existing IMS, VSAM, or sequential file.

IBM File Export for z/OS gathers data from the sources you designate, applies data manipulation rules that you specify, and then generates the desired output targets. The term *target* refers to output data.



IBM File Export offers highly selective data copying capabilities designed to simplify the complex problem of populating test environments or migrating applications to new or additional environments. IBM File Export permits you to:

- Support DB2, IMS, VSAM, and sequential sources in one product
- Automatically export all or a subset of your production data for use in your test environment using highly-flexible data selection and manipulation rules
- Extract or create files or tables that demonstrate one-to-one, one-to-many, and many-to-one relationships
- Define application relationships between dissimilar file types
- Support DB2-defined and application-defined referential integrity (RI)
- Expedite data migration and simplify data consolidation. An occasion where you might want to do this is the case where Company X has acquired Company Y and they use different databases.
- Modify the content of files or tables during export, including the ability to augment field values and age related data. For example, you can use IBM File Export for z/OS to delete rows from target DB2 tables based on keys extracted from a DB2, IMS, VSAM, or sequential source.

- Protect sensitive data by reproducible "scrambling" on a field-by-field basis

IBM File Export Interfaces

As a user, you will interface to IBM File Export by means of TSO/ISPF or by creating and running batch JCL jobs. Use the TSO/ISPF panels for "fill-in-the-blanks" specification of sources and targets, and the automatic generation of IBM File Export for z/OS processing instructions (control syntax). Alternatively, you can quickly learn IBM File Export's control syntax to write IBM File Export for z/OS JCL streams from scratch. IBM File Export's control syntax is documented in Appendix A, "Creating JCL Using Control Statements," on page 203.

Sources, targets, and associated ddnames

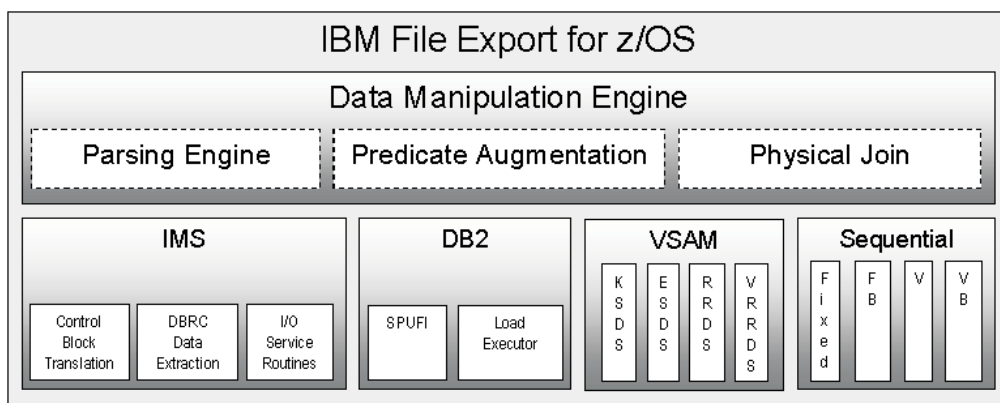
IBM File Export supports DB2, IMS (both full-function and Fast Path), VSAM, and sequential sources and targets. Easy-to-remember ddnames are used to identify the primary source, primary target, and data manipulation instructions to IBM File Export for z/OS.

- ABXIN identifies the primary source
- ABXOUT is the primary target
- ABXCTL is used to specify data gathering and manipulation instructions
- ABXPRINT identifies the data set where a copy resides of the control syntax and utility messages that are generated by IBM File Export for z/OS when it runs.

You can override the ddnames ABXIN, ABXOUT, ABXCTL, and ABXPRINT by using the appropriate parameters of the IBM File Export global options (see Appendix C, "IBM File Export Global Options," on page 277). For information on other ddnames that are used by IBM File Export for z/OS, see Appendix A, "DDnames used by IBM File Export for z/OS," on page 265.

Data manipulation

The heart of IBM File Export is the Data Manipulation Engine, ABXUTIL. Regardless of source or target file format, instructions provided to IBM File Export are processed by the Data Manipulation Engine which handles all aspects of data gathering, manipulation and generation.

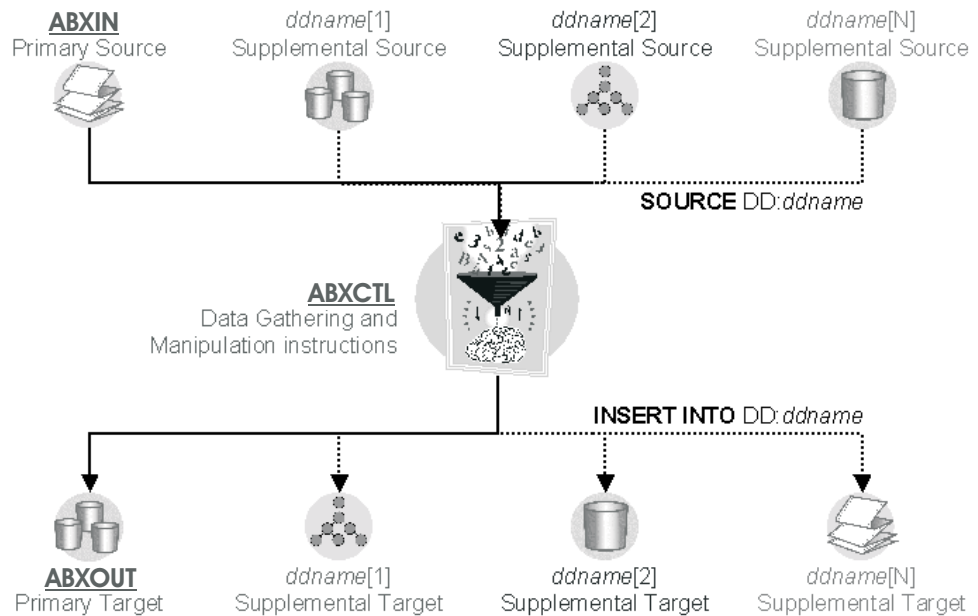


For the purpose of extracting data from sources that are not relational, IBM File Export for z/OS has drawn on the syntax of the SQL (Structured Query Language) definitions of SELECT and INSERT. IBM File Export for z/OS reads data from a

source or several sources, processes instructions which look very familiar to anyone acquainted with SQL, and then writes output to the requested target or targets.

This provides the following benefits to Application Developers and Development Test personnel:

- Ease of Use - IBM File Export for z/OS instructions are similar to SQL
- Day One Productivity - No need to learn the underlying I/O routines or spend time writing one time, "throw away" code
- Export Data On Demand - Generate IBM File Export for z/OS batch streams to automatically extract production data and load it into a test environment



The File Export repository

IBM File Export provides a repository which you can use to store information about applications and data structures that are used by more than one export. You can store data such as file layouts, record layouts, DB2 tables, and so forth. The use of the repository is optional. However, its use improves your productivity by permitting you to store information once for use by multiple applications.

Defining relationships in IBM File Export

Records can be related to each other. For example, an employee's personnel record probably contains an employee number (EMPNUM). This same employee number also exists in the payroll record, perhaps by the name EMPNO. Thus the employee's personnel record and payroll record have a relationship based on the field *employee number*. This relationship is a two-way path: `personnel.empnum = payroll.empno`. Likewise, `payroll.empno=personnel.empnum`. It is also possible that the files or tables that contain these records have been defined as having a parent-child relationship.

Relationships are specified in IBM File Export for z/OS by one of several methods:

- *Parent-child relationships* in DB2 processing .

- *Application-defined relationships* (ADRs) in DB2, VSAM, and sequential processing. Application-defined relationships are relationships that you, the user, define, rather than relationships defined by the DB2 catalog.

Application-defined relationships provide a powerful tool that permits you to create a subset of data in a *driving* table or file. You can then use this driving file to ensure that later records are extracted from a 'secondary' file only if they relate back to (match) records in the driving file.

Using the RELATIONSHIP control syntax statement, you specify an application-defined relationship by creating definitions. In these definitions, you specify a source name for data in the file, as well as the position, length, and data type of the data. A file can have as many sources for matching as desired, but each source element must have a unique name. A list of keys used for matching can be produced by previous runs of IBM File Export for z/OS using the THEN INSERT INTO dd:ddname values (captured-key) statement, or content in existing files can be used. IBM File Export's control statement syntax is documented in the chapter entitled "Creating JCL Using Control Statements" in the *IBM File Export for z/OS: Reference Manual*. A further example of application-defined relationships can be found in the same manual in the Appendix documenting the IBM File Export Installation Verification Program (IVP) for DB2.

During VSAM and sequential processing, the application-defined relationship matches content that is found in this secondary file. In DB2 processing, both application-defined relationship matching and parent-child matching can be used.

IBM File Export information

The IBM File Export library comprises the following manuals:

- *IBM File Export for z/OS: User's Guide*. This manual assists you in getting started with the product by providing a high-level overview of the IBM File Export user interface and examples of IBM File Export's use. It also provides directions for obtaining, installing, and using example code stored in the IBM File Export sample library, SABXSAMP, to further assist you with understanding the product's functions.
- *IBM File Export for z/OS: Reference Manual*. This manual documents the panels that comprise the user interface, describing the panel's fields and the valid entries you supply to define your export requests. It also documents the control statement syntax provided with ABXUTIL, the program by which you can define control statements to create JCL directly without using the product's ISPF user interface.
- *IBM File Export for z/OS: Messages and Codes*. This manual documents the messages produced by the product.

In addition, a Help panel is provided for each ISPF product panel. These Help panels provide high-level help for the fields on the panels.

Part 2. Installation and Customization

Chapter 2. Installing IBM File Export for z/OS .. 9	
Hardware requirements .. 9	
Software requirements .. 9	
Installing IBM File Export for z/OS .. 9	
Verifying that your installation is complete .. 10	
Security .. 10	
Chapter 3. Customization Tasks .. 11	
Customizing the IBM File Export CLISTs .. 11	
CLIST Member ABX .. 11	
Contents of ABX .. 11	
CLIST member ABXCLIST .. 12	
Contents of ABXCLIST .. 12	
Customizing the Global Options Block .. 13	
Security .. 13	
Changing Global Options .. 13	
Modifying CSECT ABXOGBL0 .. 13	
Customizing DB2 for use with IBM File Export .. 14	
Security .. 14	
Creating a DB2 Options Block .. 14	
Modifying CSECT ABXODB20 .. 15	
Customizing DB2 BIND jobs .. 16	
Job ABXBIND .. 16	
Contents of ABXBIND .. 16	
Job ABXBIND3 .. 17	
Contents of ABXBIND3 .. 18	
Customizing IMS for use with IBM File Export .. 20	
Security .. 20	
Processing IMS data using UNLOAD files .. 20	
Processing IMS data using DLI, DBB, or BMP regions using static PSBs .. 20	
Processing IMS data using DLI, DBB, or BMP regions using dynamic PSBs .. 20	
Creating an IMS Options Block .. 21	
Modifying CSECT ABXOIMS0 .. 21	
Customizing IBM File Export to use auditing .. 22	
Modifying an ISPF panel to start IBM File Export .. 23	

This part contains information on installing IBM File Export for z/OS and verifying that the installation is complete. Once these tasks are done, you can log onto IBM File Export and select the Setup option to configure the system to your specific requirements.

Chapter 2. Installing IBM File Export for z/OS

This chapter contains information about the hardware and software required to install IBM File Export for z/OS, as well as information about installing the product in your environment. Once the installation is complete, you can log onto IBM File Export to configure the system for your specific requirements.

Hardware requirements

To successfully run IBM File Export, you must have a zSeries® machine capable of running the required software as listed in “Software requirements”.

Software requirements

The following software is required to use IBM File Export for z/OS:

Host system:

- z/OS® V1R4 or later
- XML ToolkitV1R6M0

IMS support:

- To process data from IMS databases, either of:
 - IMS Database Manager V07.01.00 (5655-B01)
 - IMS Database Manager V08.01.00 (5655-C56)
 - IMS Database Manager V09.01.00 (5655-J38)

IMS Fast-Path DEDB data support:

To process data from IMS Fast-Path DEDB databases:

- IMS Fast-Path Basic Tools for z/OS V01.02.00 (5655-F78) or later

DB2 support:

- IBM DB2 for z/OS V6 or DB2 V7
- IBM DB2 for z/OS V8 in compatibility mode
- IBM DB2 Utilities Suite for z/OS and S/390

Installing IBM File Export for z/OS

1. Retrieve these binary files from the distribution medium to your host machine:
 - SABXDBRM
 - SABXMENU
 - SABXPENU
 - SABXSLIB
 - SABXLOAD
 - SABXSAMP

These files must be binary, and may not be an EBCDIC translation. They must not contain carriage returns or line feeds. These files were all generated using TSO XMIT and will be received using TSO RECEIVE (see step 2 below).

Notes:

- a. The suggested high level qualifier is 'ABX'
 - b. The suggested data set names are:
 - ABX.V110.SABXDBRM
 - ABX.V110.SABXMENU
 - ABX.V110.SABXPENU
 - ABX.V110.SABXSLIB
 - ABX.V110.SABXLOAD -- ABX.V110.SABXLOAD must be APF-authorized.
 - ABX.V110.SABXSAMP
2. Issue TSO RECEIVE for each file. For example, from the READY prompt:
RECEIVE INDSNAME('SABXLOAD')
Type the following at the prompt:
DATASET('ABX.V110.SABXLOAD')
 3. Bind the ABX plans. Customize and submit member ABXBIND of the SABXSAMP data set.
 4. Run this bind job for each subsystem that you want to access.
 5. Ensure that all plans have been granted sufficient authority.
 6. Setup the defaults for the product.
 - Copy the member ABX from the SABXSAMP data set to a CLIST library that is available through the TSO user's SYSPROC concatenation
 - Review/Edit the ABX member and update the HILEVEL variable to reflect the actual Hilevel qualifier used at the site if other than ABX.V110
 - Start the product by running the ABX CLIST.
 - Configure IBM File Export for z/OS. For more information on this see "Configuring IBM File Export global options" on page 29.

Verifying that your installation is complete

Once you have completed installing and configuring IBM File Export for z/OS, you will want to run the appropriate Installation Verification Program (IVP). The IVPs that are available to you are described in Appendix B, "Installation Verification Programs," on page 273.

Security

All IBM File Export operations create SMF records for auditing purposes. Security will be managed by standard RACF security with the following exception:

- Where dynamic PSBs are created and used, security exits will be called to allow you to ensure that installation security policies are not violated.

Chapter 3. Customization Tasks

This chapter discusses some of the customization tasks that must be accomplished to successfully run IBM File Export for z/OS.

Customizing the IBM File Export CLISTS

To start the IBM File Export user interface, you run the IBM File Export CLISTS. Before doing so, you must first customize the CLISTS ABX and ABXCLIST, located in the SABXSAMP sample library, to work in your environment. Then, copy the customized CLISTS from SABXSAMP to a CLIST library that is in all users' SYSPROC concatenation for CLISTS, or modify each user's logon procedure to include ABX.SABXSAMP.

<p>Note: When customizing the various parameters in a CLIST, you must ensure that the case of the values you enter must exactly match the case of the corresponding information in your Hierarchical File System (HFS) directory.</p>
--

CLIST Member ABX

The statements comprising member ABX execute the contents of member ABXCLIST. The statements within member ABX allow you to start the IBM File Export user interface CLIST, ABXCLIST, as a separate application. This enables you to continue your ISPF session should you encounter a problem with this particular instance of IBM File Export. The contents of the IBM File Export CLIST member ABX are shown below. You should note that this CLIST as distributed with the product is the most current version and may differ slightly from the listing presented here.

To customize member ABX, perform the following steps:

1. Copy the ABX and ABXCLIST members to your system CLIST library. You must do so in order to execute the ABX CLIST using TSO.
2. In the `PROC 0 CLISTLIB(name)` statement, specify the name of the data set to which the ABXCLIST member was copied. For example,
`PROC 0 CLISTLIB(ABCD.USER.CLIST)`

Contents of ABX

```
PROC 0 CLISTLIB()
/* LICENSED MATERIALS - PROPERTY OF IBM
/* 5697-I12
/* (C) COPYRIGHT IBM CORP. 2003, 2004 ALL RIGHTS RESERVED.
/* (C) COPYRIGHT ROCKET SOFTWARE, INC. 2003-2004 ALL RIGHTS RESERVED.
/* US GOVERNMENT USERS RESTRICTED RIGHTS - USE, DUPLICATION OR
/* DISCLOSURE RESTRICTED BY GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
CONTROL NOMSG
```

```

FREE FILE(ABXCLIST)

CONTROL MSG

ALLOC FILE(ABXCLIST) DATASET('&CLISTLIB') SHR REU

ALTLIB ACTIVATE APPLICATION(CLIST) FILE(ABXCLIST)

ISPEXEC SELECT CMD(ABXCLIST) NEWAPPL(ABX) PASSLIB

ALTLIB DEACTIVATE APPLICATION(CLIST)

FREE FILE(ABXCLIST)

```

CLIST member ABXCLIST

ABXCLIST starts the IBM File Export user interface. The contents of ABXCLIST are shown below. You should note that this CLIST as distributed with the product is the most current version and may differ slightly from the listing presented here. You need to ensure that any data set names after variable substitution do not exceed 44 characters in length. You must also customize ABXCLIST by making the following modifications:

1. In the first line of ABXCLIST, set the variable HILEVEL to the high level qualifier of the data set where you have installed IBM File Export
2. Modify the second line of ABXCLIST to point to the data set that contains the XML Toolkit Load Library.

Contents of ABXCLIST

```

PROC 0 HILEVEL(HILEVEL) +
      XMLDSN(XMLDSN)

/* LICENSED MATERIALS - PROPERTY OF IBM
/* 5697-I12
/* (C) COPYRIGHT IBM CORP. 2003, 2004 ALL RIGHTS RESERVED.
/* (C) COPYRIGHT ROCKET SOFTWARE, INC. 2003-2004 ALL RIGHTS RESERVED.
/* US GOVERNMENT USERS RESTRICTED RIGHTS - USE, DUPLICATION OR
/* DISCLOSURE RESTRICTED BY GSA ADP SCHEDULE CONTRACT WITH IBM CORP.

CONTROL NOMSG

FREE FILE(ABXUTIL)

CONTROL MSG

SET &ABXLIB = &STR(&HILEVEL..SABXLOAD)

ISPEXEC LIBDEF ISPMLIB DATASET ID('&HILEVEL..SABXMENU') UNCOND
ISPEXEC LIBDEF ISPLLIB DATASET ID('&HILEVEL..SABXPENU') UNCOND
ISPEXEC LIBDEF ISPSLIB DATASET ID('&HILEVEL..SABXSLIB') UNCOND
ISPEXEC LIBDEF ISPTLIB DATASET ID('&HILEVEL..SABXTENU') UNCOND
ISPEXEC LIBDEF ISPLLIB DATASET ID('&ABXLIB' +
                                '&XMLDSN') UNCOND

```

```
ALLOC F(ABXUTIL) DSN('&ABXLIB') SHR
ABXMAIN &ABXLIB
FREE FILE(ABXUTIL)
```

Customizing the Global Options Block

In order to take full advantage of the features of IBM File Export and simplify its use, we recommend that you perform some customization. This customization process must take place after the product has been installed in your environment. The process should be performed by someone who has UPDATE authority to the product data sets.

Security

Security for the basic IBM File Export product, which includes Sequential and VSAM processing, is provided by RACF's normal data set protection, or that of a similar security product. No IBM File Export-specific customization is required to protect VSAM or Sequential data sets. For DB2 or IMS security information, see the sections entitled "Security" on page 14 for DB2 and on page 20 for IMS.

Changing Global Options

IBM File Export permits you to change a number of default processing options to values specific to your installation. In order to effect these changes, you must create a LOAD module named ABXOGBL0 that must reside in the IBM File Export program product library. To create this LOAD module, an Assembler macro, ABXMGBL0, and a CSECT, ABXOGBL0, along with assembly and link-edit JCL, ABXBLDGB, are supplied with IBM File Export. These members are found in the sample library dataset ABX.V110.SABXSAMP.

You can modify most of the options available for individual executions by using control statement keywords. See Appendix C, "IBM File Export Global Options," on page 277 and Chapter 10, "Creating JCL Using Control Statements," on page 203 for additional information.

Modifying CSECT ABXOGBL0

This section provides a listing of CSECT ABXOGBL0 and illustrates the use of macro ABXMGBL0. The supplied default values are illustrated. To change global options using this CSECT, you must:

1. Copy member ABXOGBL0 into your own source library
2. Modify the options in the ABXOGBL0 member of your library as needed.
3. Copy member ABXBLDGB into your own JCL library.
4. Modify the data set names in the ABXBLDGB member of your library to reflect your input and output data set names.
5. Submit the ABXBLDGB member of your library for execution.

ABXOGBLO TITLE 'File Export for z/OS : Global Processing Options'

```
*****
* Licensed Materials - Property of IBM
* 5697-I12
* (C) Copyright IBM Corp. 2003 - 2004 All Rights reserved.
* (C) Copyright Rocket Software 2003 - 2004 All Rights Reserved
* US Government Users Restricted Rights - Use, duplication
* or disclosure restricted by GSA ADP Schedule Contract
* with IBM Corp.
*
*
*Function = Default options information.
* ABXOGBLO is used to set the default processing options
* for IBM File Export base function.
*
* For information on coding the options module ABXOGBLO
* refer to the Installation and Customization sections of
* this IBM File Export for z/OS Reference Manual.
*
*****
* ABXMGBLO
+
+ AUDIT=NO, or YES or FORCE +
+ INDDN=ABXIN, DD name of input dataset +
+ OTDDN=ABXOUT, DD name of output dataset +
+ PTDDN=ABXPRINT, DD name of messages dataset +
+ SINDDN=ABXCTL, DD name of control statements +
+ WUNIT=SYSDA Work device esoteric name
*
* END ABXOGBLO
```

You can also modify some of the product's global options using the Settings (ABX\$GOPT) panel, available from the IBM File Export ISPF user interface. See the section entitled "Configuring IBM File Export global options" on page 29 for information about those global options available from the Settings panel.

Customizing DB2 for use with IBM File Export

Processing DB2 data using IBM File Export can be made more efficient and simpler for the end user by performing some customization. This customization process must take place after the product has been installed in your environment. The process should be performed by someone who has UPDATE authority to the IBM File Export product data sets and is able to issue DB2 GRANT commands.

Security

Security against accidental or unauthorized access to DB2 tables is provided using normal DB2 security facilities.

Creating a DB2 Options Block

IBM File Export permits you to generate a single option block that contains the default execution options for multiple DB2 subsystems. In order to supply these default values, you must create a LOAD module named ABXODB20 that must reside in the IBM File Export program product library. To create this LOAD module, an Assembler macro, ABXMDB20, and a CSECT, ABXODB20, along with assembly and link-edit JCL, ABXBLDB2, is supplied with IBM File Export. These members are found in the sample library dataset ABX.V110.SABXSAMP.

You can modify most of the options available for individual executions by using control statement keywords. See Appendix D, "IBM File Export DB2 Options," on page 281 and Chapter 10, "Creating JCL Using Control Statements," on page 203 for additional information.

Modifying CSECT ABXODB20

This section provides a listing of CSECT ABXODB20 and illustrates the use of macro ABXMDB20. To change DB2 options using this CSECT, you must:

1. Copy member ABXODB20 into your own source library
2. Add an instance of ABXMDB20 TYPE=ENTRY for each DB2 subsystem that will be supported by IBM File Export.
3. Modify each TYPE=ENTRY instance to reflect the options for that specific DB2 subsystem.
4. Copy member ABXBLDB2 into your own JCL library.
5. Modify the data set names in the ABXBLDB2 member of your library to reflect your input and output data set names.
6. Submit the ABXBLDB2 member of your library for execution.

```

ABXODB20 TITLE 'IBM File Export for z/OS : DB2 Processing Options'
*****
*      Licensed Materials - Property of IBM
*      5697-I12
*      (C) Copyright IBM Corp. 2003 - 2004 All Rights reserved.
*      (C) Copyright Rocket Software 2003 - 2004 All Rights Reserved
*      US Government Users Restricted Rights - Use, duplication
*      or disclosure restricted by GSA ADP Schedule Contract
*      with IBM Corp.
*
*
*      Function = Default options information.
*      ABXODB20 is used to set the default processing options
*      for IBM File Export DB2 function.
*
*      The ABXMDB20 TYPE=START must precede any TYPE=ENTRY
*      statements.
*
*      The ABXMDB20 TYPE=FINAL must be the last coded ABXMDB20
*      macro.
*
*      For information on coding the options module ABXODB20
*      refer to the Installation and Customization sections
*      of this IBM File Export for z/OS Reference Manual.
*****
*
*****
** Indicate/initialize the start of entry processing      **
*****
          ABXMDB20 TYPE=START
*
          ABXMDB20 TYPE=ENTRY,SSID=db2id,LOADLIB=db2.loadlib
*
*****
** Indicate that all TYPE=ENTRY instances have been coded      **
*****
          ABXMDB20 TYPE=FINAL
          END  ABXODB20

```

Customizing DB2 BIND jobs

If you are using the DB2 facilities of IBM File Export, you need to customize one of the two DB2 BIND jobs, ABXBIND or ABXBIND3, to meet your installation requirements. These jobs can be found in SABXSAMP.

Once you have customized the appropriate BIND job, you must execute that job against each DB2 subsystem that you plan to use with IBM File Export for z/OS. You will need to let your users know the plan name(s) that you selected. You also need to ensure that any data set names after variable substitution do not exceed 44 characters in length.

Job ABXBIND

Job ABXBIND provides an example of a DB2 BIND and GRANT to allow DB2 access using IBM File Export for z/OS. The job binds all of the IBM File Export DBRM modules into a single plan. If you want to bind each DBRM separately, you should use job ABXBIND3. You may need to modify this job for your environment and your installed product set. The example code is meant to be illustrative only.

Contents of ABXBIND

```
//ABXBIND JOB 'BIND ABXUTIL',CLASS=A,MSGCLASS=X,
//          NOTIFY=&SYSUID
//*
//*****
/* Licensed Materials - Property of IBM
/* 5697-I12
/* (c) Copyright IBM Corp. 2003, 2004 All Rights Reserved.
/* (c) Copyright Rocket Software, Inc. 2003 - 2004 All Rights Reserve
/* US Government Users Restricted Rights -
/* Use, duplication or Disclosure restricted by GSA ADP
/* Schedule Contract with IBM Corp.
//*****/
/*
//*****
/* This job gives an example of a DB2 BIND and GRANT to allow DB2 *
/* access using IBM File Export for z/OS. *
/* *
/* This job binds all of the File Export DBRM modules into a single *
/* plan. If you wish to bind each DBRM separately, you should use*
/* job ABXBIND3. *
/* *
/* You may need to make modifications for your environment and your *
/* installed product set. This example code is meant to be *
/* illustrative only and has not been thoroughly tested under *
/* all possible operating conditions. *
/* *
//*****
//*
```

```

//*****
/* THIS JOB MUST BE RUN FOR EACH DB2 THAT WILL BE ACCESSED WITH      *
/* IBM FILE EXPORT FOR Z/OS                                          *
/*                                                                    *
/*                                                                    *
/* THE FOLLOWING CHANGES MUST BE MADE TO THIS JOB BEFORE SUBMITTING:*
/*                                                                    *
/* 1. CHANGE "DB2.LOADLIB" TO THE NAME OF YOUR DB2 LOAD LIBRARY      *
/* 2. CHANGE "DB2SUBSYS" TO THE SUBSYSTEM ID OF YOUR DB2 SUBSYSTEM  *
/* 3. CHANGE "PLANOWNER" TO THE SITE SPECIFIC OWNER                  *
/* 4. CHANGE "ABXHLQ.SABXDBRM" TO THE NAME OF YOUR FILE EXPORT      *
/*    DBRM LIBRARY                                                  *
/*                                                                    *
/* 5. IF YOU DO NOT WISH TO USE THE GRANT BELOW, YOU MAY COMMENT IT *
/*    OUT.                                                            *
/*                                                                    *
/*                                                                    *
//*****
/*
//ABXUTIL EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB DD DISP=SHR,DSN=DB2.LOADLIB
//SYSTSIN DD *
    DSN SYSTEM(DB2SUBSYS)
    BIND PLAN (ABXPLAN) -
        MEMBER (ABXUTIL -
            ABXDBUT -
            ABXD2LEX) -
        ACTION (REPLACE) -
        RETAIN -
        EXPLAIN (NO) -
        ISOLATION (CS) -
        SQLRULES (DB2) -
        OWNER(PLANOWNER)
//SYSIN DD *
    GRANT USE ON PLAN ABXPLAN TO PUBLIC
/*
//DBRMLIB DD DISP=SHR,DSN=ABXHLQ.SABXDBRM
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*

```

Job ABXBIND3

Job ABXBIND3 provides an example of a DB2 BIND and GRANT to allow DB2 access using IBM File Export for z/OS. The job binds the IBM File Export DBRM modules into three separate plans. If you want to bind all of the DBRMs into a single plan, you should use job ABXBIND. You may need to modify this job for your

environment and your installed product set. The example code is meant to be illustrative only.

Contents of ABXBIND3

```
//ABXBIND3 JOB 'BIND ABXUTIL',CLASS=A,MSGCLASS=X,
//          NOTIFY=&SYSUID
//*
//*****
/* Licensed Materials - Property of IBM
/* 5697-I12
/* (c) Copyright IBM Corp. 2003, 2004 All Rights Reserved.
/* (c) Copyright Rocket Software, Inc. 2003 - 2004 All Rights Reserve
/* US Government Users Restricted Rights -
/* Use, duplication or Disclosure restricted by GSA ADP
/* Schedule Contract with IBM Corp.
//*****/
/*
//*****
/* This job gives an example of a DB2 BIND and GRANT to allow DB2 *
/* access using File Export for z/OS. *
/* *
/* This job binds the File Export DBRM modules into three separate *
/* plans. If you wish to bind the DBRMs into a single plan, you *
/* should use job ABXBIND. *
/* *
/* You may need to make modifications for your environment and your *
/* installed product set. This example code is meant to be *
/* illustrative only and has not been thoroughly tested under *
/* all possible operating conditions. *
/* *
//*****
/*
//*****
/* THIS JOB MUST BE RUN FOR EACH DB2 THAT WILL BE ACCESSED WITH *
/* IBM FILE EXPORT FOR Z/OS *
/* *
/* *
/* THE FOLLOWING CHANGES MUST BE MADE TO THIS JOB BEFORE SUBMITTING:*
/* *
/* 1. CHANGE "DB2.LOADLIB" TO THE NAME OF YOUR DB2 LOAD LIBRARY *
/* 2. CHANGE "DB2SUBSYS" TO THE SUBSYSTEM ID OF YOUR DB2 SUBSYSTEM *
/* 3. CHANGE "PLANOWNER" TO THE SITE SPECIFIC OWNER *
/* 4. CHANGE "ABXHLQ.SABXDORM" TO THE NAME OF YOUR FILE EXPORT *
/* DBRM LIBRARY *
/* *
```



```

/* 5. IF YOU DO NOT WISH TO USE THE GRANTS BELOW, YOU MAY COMMENT *
/* THEM OUT. *
/* *
/* *
/******
/*
/*ABXUTIL EXEC PGM=IKJEFT01,DYNAMNBR=20
/*STEPLIB DD DISP=SHR,DSN=DB2.LOADLIB
/*SYSTSIN DD *
    DSN SYSTEM(DB2SUBSYS)
    BIND PLAN (ABXUTIL) -
        MEMBER (ABXUTIL) -
        ACTION (REPLACE) -
        RETAIN -
        EXPLAIN (NO) -
        ISOLATION (CS) -
        SQLRULES (DB2) -
        OWNER(PLANOWNER)

    BIND PLAN (ABXD2UI) -
        MEMBER (ABXDBUT) -
        ACTION (REPLACE) -
        RETAIN -
        EXPLAIN (NO) -
        ISOLATION (CS) -
        SQLRULES (DB2) -
        OWNER(PLANOWNER)

    BIND PLAN (ABXD2LEX) -
        MEMBER (ABXD2LEX) -
        ACTION (REPLACE) -
        RETAIN -
        EXPLAIN (NO) -
        ISOLATION (CS) -
        SQLRULES (DB2) -
        REOPT (VARS) -
        OWNER(PLANOWNER)
/*SYSDIN DD *
    GRANT USE ON PLAN ABXUTIL TO PUBLIC
    GRANT USE ON PLAN ABXD2UI TO PUBLIC
    GRANT USE ON PLAN ABXD2LEX TO PUBLIC
/*
/*DBRMLIB DD DISP=SHR,DSN=ABXHLQ.SABXDBRM
/*SYSPPRINT DD SYSOUT=*
/*SYSTSPRT DD SYSOUT=*

```

Customizing IMS for use with IBM File Export

Processing IMS data using IBM File Export can be made more efficient and simpler for the end user by performing some customization. This customization process must take place after the product has been installed in your environment. The process should be performed by someone who has UPDATE authority to the IBM File Export product data sets and is able to issue RACF or similar security product commands.

Security

Security against accidental or unauthorized access to IMS databases is provided by IBM File Export, which uses RACF or a similar security product.

Processing IMS data using UNLOAD files

Permission to process IMS UNLOAD files is controlled by a security product's normal data set protection, since the UNLOAD files are sequential files processed using standard QSAM I/O. No IBM File Export-specific customization is required.

Processing IMS data using DLI, DBB, or BMP regions using static PSBs

No IBM File Export-specific customization to prevent accidental or unauthorized access to IMS databases is required when using static PSBs, since IBM File Export uses standard IMS batch services to read and write IMS data in these processing modes. To learn more about static and dynamic PSBs, see the chapter so entitled in the *IBM File Export for z/OS: User's Guide*.

Processing IMS data using DLI, DBB, or BMP regions using dynamic PSBs

In order to utilize the Dynamic PSB function of IBM File Export, some customization is required to restrict database access to authorized users. It should be noted that the use of the IMS Get Unique (GU) function of IBM File Export *requires* the use of dynamic PSBs. If you will not use the IMS GU function or, in general, do not use dynamic PSBs at your installation, this step need not be followed. If you decide later to utilize the Dynamic PSB function, you can perform this procedure without impacting any other installation steps.

To protect against the unauthorized use of the Dynamic PSB function, IBM File Export uses Security Server, RACF 1.9 or later, or equivalent security product, and the System Authorization Facility (SAF). IBM File Export requires that the FACILITY class be active on your system. To activate the FACILITY class on your system, issue the following security program commands:

```
SETROPTS CLASSACT(FACILITY)
SETROPTS GENERIC(FACILITY)
SETROPTS GENCMD(FACILITY)
```

IBM File Export requires that the facility name ABX.DYNPSB.ALL be defined to your system. To define the FACILITY name to your system, issue the following security program commands:

```
REDEFINE FACILITY ABX.DYNPSB.ALL UACC(NONE)
SETROPT RACLIST (FACILITY) REFRESH
```

Note: The REDEFINE and SETROPT commands may vary in form in your installation from the example above.

We suggest that you create an access group that has READ permission to the ABX.DYNPSB.ALL facility. This permits users who are connected to this group to execute jobs that create dynamic PSBs that are only capable of reading IMS databases. Also, creating a group that has UPDATE permission to the ABX.DYNPSB.ALL facility permits connected users to execute jobs that create dynamic PSBs that may read and update IMS databases.

To learn more about static and dynamic PSBs, see the chapter so entitled in the *IBM File Export for z/OS: User's Guide*

Creating an IMS Options Block

IBM File Export permits you to generate a single option block that contains the default execution options for multiple IMS subsystems. In order to supply these default values, you must create a LOAD module named ABXOIMS0 that must reside in the IBM File Export program product library. To create this LOAD module, an Assembler macro, ABXMIMS0, and a CSECT, ABXOIMS0, along with assembly and link-edit JCL, ABXBLDIM, is supplied with IBM File Export. These members are found in the sample library dataset ABX.V110.SABXSAMP.

You can modify most of the options available for individual executions by using control statement keywords (see Appendix E, "IBM File Export IMS Options," on page 283 and Chapter 10, "Creating JCL Using Control Statements," on page 203 for additional information).

Modifying CSECT ABXOIMS0

This section provides a listing of CSECT ABXOIMS0 and illustrates the use of macro ABXMIMS0. To change IMS options using this CSECT, you must:

1. Copy member ABXOIMS0 into your own source library
2. Add an instance of ABXMIMS0 TYPE=ENTRY for each IMS subsystem that will be supported by IBM File Export.
3. Modify each TYPE=ENTRY instance to reflect the options for that specific IMS subsystem.
4. Copy member ABXBLDIM into your own JCL library.
5. Modify the data set names in the ABXBLDIM member of your library to reflect your input and output data set names.
6. Submit the ABXBLDIM member of your library for execution.

```

ABXOIMS0 TITLE 'File Export for z/OS : IMS Processing Options'
*****
* Licensed Materials - Property of IBM
* 5697-I12
* (C) Copyright IBM Corp. 2003 - 2004 All Rights reserved.
* (C) Copyright Rocket Software 2003 - 2004 All Rights Reserved
* US Government Users Restricted Rights - Use, duplication
* or disclosure restricted by GSA ADP Schedule Contract
* with IBM Corp.
*
*
* Function = Default options information.
* ABXOIMS0 is used to set the default processing options
* for File Export IMS function.
*
* The ABXMIMS0 TYPE=START must precede any TYPE=ENTRY
* statements.
*
* The ABXMIMS0 TYPE=FINAL must be the last coded ABXMIMS0
* macro.
*
* For information on coding the options module ABXOIMS0
* refer to the Installation and Customization sections
* of this File Export for z/OS Reference Manual.
*
*****
*
*****
** Indicate/initialize the start of entry processing
**
*****
ABXMIMS0 TYPE=START
*
ABXMIMS0 TYPE=ENTRY,IMSID=SSID
*
*****
** Indicate that all TYPE=ENTRY instances have been coded **
*****
ABXMIMS0 TYPE=FINAL
END ABXOIMS0

```

Customizing IBM File Export to use auditing

You can use the audit logging facilities of IBM File Export to create an audit trail of IBM File Export activity. Your audit is written to SMF, from which you can later extract and print a report. Set your audit options in the Global Options block. To use the audit function, you must also:

- Ensure that the load module ABXUTIL is APF-authorized by authorizing the LOAD library.
- Select an SMF record number between 128 and 255 inclusive for audit records and include this number in your SMF PARMLIB member SMFPRFxx.
- Specify this SMF record number in ABXOGLB0.
- Specify **YES** or **FORCE** for the AUDIT= parameter of the ABXMGLB0 macro contained in the ABXOGLB0 Global Control block.

To report on the audit trail information collected by SMF, you must extract this information from SMF to your own data set. The information in this data set can then be printed by the IBM File Export Print Audit facility, ABXPAUD. A sample job, ABXSPAUD, is provided in ABX.SABXSAMP to help you extract the SMF data to your own data set. The comments in the job direct you in making required changes to the job.

You can also specify audit options using the Settings (ABX\$GOPT) panel that is available from the IBM File Export ISPF user interface. See the section entitled “Configuring IBM File Export global options” on page 29 for information about the Settings panel.

For additional information about SMF, see the IBM manual entitled *z/OS MVS : System Management Facilities (SMF)*, SA22-7630.

Modifying an ISPF panel to start IBM File Export

If you wish to start IBM File Export other than by invoking the ABX CLIST under ISPF, you should modify an installation-specific ISPF panel to add a suitable option that, when selected, invokes this CLIST.

Part 3. Start-up and Configuration

Chapter 4. Start-up and Configuration using the IBM File Export User Interface	27	Selecting and managing your IBM File Export sessions	28
The IBM File Export Main Menu	27	Configuring IBM File Export global options	29
Administering the IBM File Export repository ...	28		

This part walks you through the user interface as you begin to use IBM File Export. It shows you the functions available to you from IBM File Export Main Menu and assists you in configuring your product using the Global and DB2 product option panels.

Chapter 4. Start-up and Configuration using the IBM File Export User Interface

Once your installation is complete, you will want to verify that the product has been installed correctly by using an Installation Verification Program as described in Appendix B, “Installation Verification Programs,” on page 273. The next step is to configure your particular installation using the IBM File Export Settings option. You will need to define global product settings. If you will use DB2, you will also need to run the DB2 setup as described in “Specify DB2 product options” on page 105.

The IBM File Export Main Menu

From the IBM File Export Main Menu, you can:

- Choose to administer your IBM File Export repository
- Select Session Management to create a new IBM File Export session or edit or use an existing session
- Define global settings for IBM File Export

You can also use this panel to display a pop-up window showing information about the currently installed product, or you can exit the product. To proceed:

1. Start the product by running the ABX CLIST (see “Customizing the IBM File Export CLISTs” on page 11) or from a modified ISPF panel (see “Modifying an ISPF panel to start IBM File Export” on page 23).
2. Select the appropriate option by typing its number on the Option line.

```
ABX$MAIN      -----  IBM File Export for z/OS  -----  2004/09/17  10:04:15
Option  ==>

1  Repository Management
2  Session Management

S  Settings
A  About
X  Exit
```

Figure 1. IBM File Export Main Menu (ABX\$MAIN)

Panel fields:

1. Specifying **1** on the Option line displays the Repository Management panel, from which you can manage your IBM File Export repository.
2. Specifying **2** on the Option line displays the Session Management panel, from which you can create a new session, open an existing session, or view a list of all existing sessions.

- S Specifying **S** on the Option line displays the Settings panel, where you can define settings for the product.
- A Specifying **A** on the Option line displays a pop-up window that provides information about the current release of IBM File Export.
- X Specifying **X** on the Option line permits you to exit IBM File Export.

Administering the IBM File Export repository

Selecting Option 1 on the Main Menu displays the Repository Management panel. From this panel, you can create, delete, or open the repository.

```
ABX$RMNG      ----- Repository Management ----- 2004/09/17 10:02:50
Option ==> 1

1 Create
2 Delete
3 Open
```

Figure 2. IBM File Export Repository Management (ABX\$RMNG) panel

Panel fields:

- 1 Specifying **1** permits you to create the repository.
- 2 Specifying **2** permits you to delete the repository.
- 3 Specifying **3** permits you to open the repository.

See Chapter 5, “Working with the IBM File Export Repository,” on page 35 for information about using the IBM File Export repository functions.

Selecting and managing your IBM File Export sessions

From the Session Management (ABX\$SMNG) panel, you can choose the type of session you want to create, open an existing session, or view a list of existing sessions from which you can select the session to work with. The Session Management panel is discussed in detail in Chapter 6, “Getting Started with the IBM File Export ISPF Interface,” on page 101.

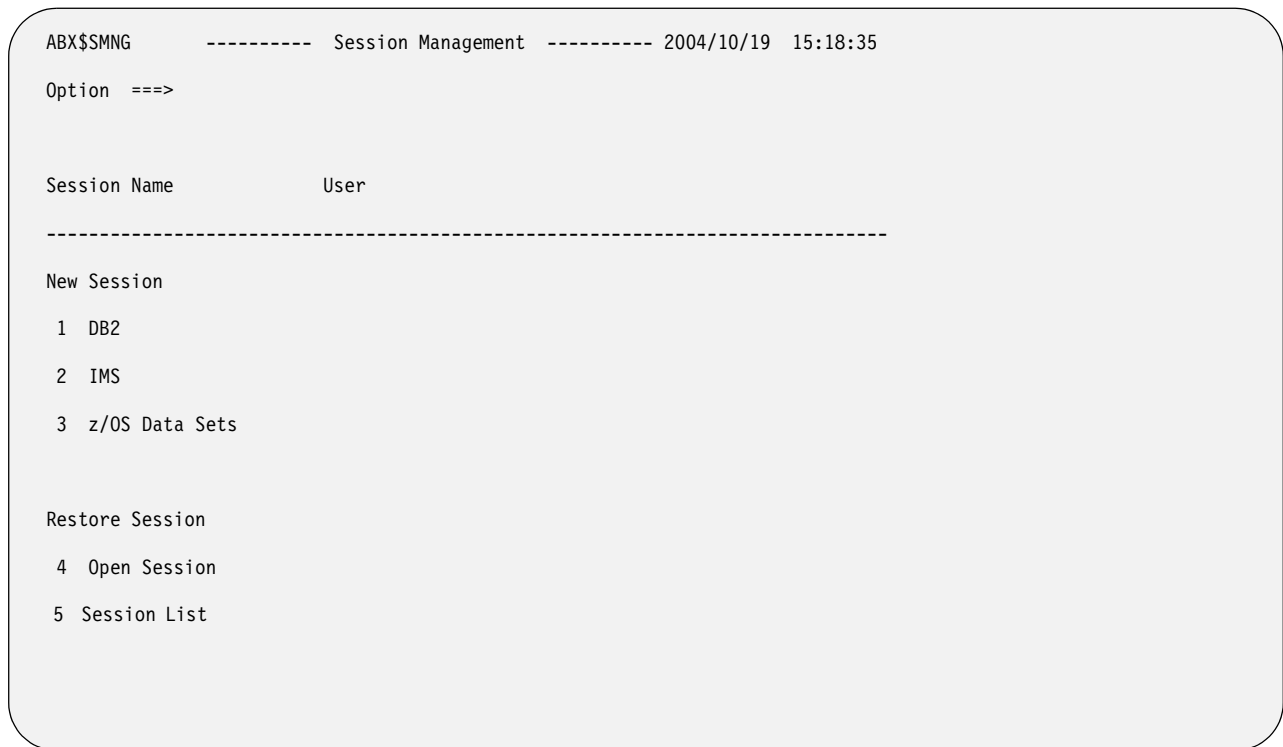


Figure 3. IBM File Export Session Management (ABX\$SMNG) panel

Panel fields:

- 1 Specifying **1** starts a new DB2 session. You must also specify a session name in the **Session Name** field.
- 2 Specifying **2** starts a new IMS session. You must also specify a session name in the **Session Name** field.
- 3 Specifying **3** starts a new sequential or VSAM session. You must also specify a session name in the **Session Name** field.
- 4 Specifying **4** starts an existing session whose name you have specified in the **Session Name** field.
- 5 Specifying **5** displays list of existing sessions. If you have included anything in the **Session Name** field, it is ignored.

Configuring IBM File Export global options

Use the Settings panel to configure some of IBM File Export for z/OS's global options. Note that some global options cannot be modified through the user interface. For information, see "Customizing the Global Options Block" on page 13.

1. From the Main Menu (ABX\$MAIN) panel, select Option S (Settings).
2. The Settings (ABX\$GOPT) panel opens.

```

ABX$GOPT ----- Settings ----- 2004/10/28 19:08
Option ==>

More:

Work Unit      SYSDA      Work Volume
Fixed Block    Quiet          Y (Yes / No)
High Level Qual      Field          (Yes / No)
User Exit      (Yes / No)    User Exit Prgm

Audit          (Yes / No / Force) SMF Record

Repository HLQ MDSTIT.ABX.R1
JCL Library
Session Library MDSTIT.ABX.STORLIB
Work DSN       ABX.EDITWORK

Job Statement Information:
==> //JOB CARD JOB MDSTIT,CLASS=A,NOTIFY=&SYSUID
==> /*
==> /*
==> /*

Enter D command to setup DB2 options now, or PF3 to exit

```

Figure 4. Settings ABX\$GOPT) Menu

Panel fields:

Fill Character

Type the character that you want to use for padding in IBM File Export for z/OS in this field.

Decimal

Type **Decimal** or **Comma** to specify the character to be used as the decimal point.

Sysout Class

Type the output class that you want to use for JCL in this field.

Wild Card Char

Type the character that you want to use as a wildcard in IBM File Export for z/OS in this field. The wild card is used in the LIKE predicate and represents the multi-character non-matching mask.

Note: In DB2 mode, wild card processing is native SQL and DB2 uses only the percent sign (%), which cannot be overridden.

Work unit

Specify the name of the unit that you want to use for work files. The default is SYSDA.

Work Volume

Type the name of the volume that you want to use for product work data sets.

Fixed Block

If you want to force a fixed length record, type the length for records in this field. This is ignored for IMS processing.

Quiet

Type **Y** in this field to suppress the printing of runtime messages normally emitted by PRINT requests on SELECT and INSERT statements as well as the printing of records that fail the UNIQUE directive. Typing **N** in this field causes these messages and records to be printed.

High Level Qual

Type the high level qualifier that you want to use to prefix IBM File Export for z/OS output data sets.

Field Type **Y** in this field to produce a message for each field, specifying its attributes. Type **N** to suppress these messages. The messages are produced when the fields from the source DBD are automatically defined in the input DBD when the DBDLIB definition is run.

User Exit

Type **Y** in this field to specify a user exit program is to be called to process I/Os. Type **N** to specify that you do not want to call a user I/O exit.

Note: If you have specified an I/O exit name using the User Exit Prgm keyword, but you have specified **No** for the User Exit keyword, no exit processing will occur. However, if you have specified **Yes** for the User Exit keyword and have not supplied a user exit program name, you will receive an error message.

User Exit Prgm

Type the name of the user exit program in this field.

Audit

Type **N** (the default) in this field to indicate that SMF audit records are not to be written to the system SMF data sets. Type **Y** in this field to indicate that SMF audit records are to be written to the system SMF data sets. If, for some reason, the request to write the SMF records fails, IBM File Export issues a message indicating the reason for the failure, dynamically changes the AUDIT option to NO, and continues processing the input and output data without an audit trail. Type **F** in this field to indicate that SMF audit records are to be written to the system SMF data sets. If, for some reason, the request to write the SMF records fails, IBM File Export issues a message indicating the reason for the failure, dynamically changes the AUDIT option to NO, and terminates processing.

SMF Record

Displays an SMF record number between 128 and 255 inclusive for audit records.

Repository HLQ

Displays the name of your default repository.

JCL Library

Type the name of the PDS member that will contain the JCL generated by IBM File Export for z/OS. You must define a JCL library in order to save the JCL you will generate using IBM File Export for z/OS.

Session Library

Type the name of the library that is to contain the information related to a specific session of IBM File Export for z/OS. If you do not specify a session library, the session you are defining will not be saved.

Work DSN

Type the name of the work data set to be used by IBM File Export for z/OS for DB2 Exports. If this field is left blank, USERID.ABX.EDITWORK will be used by default. This data set is always prefaced with your TSO userID.

Job statement information

Type the job statement information to be used by IBM File Export for z/OS. You should include a REGION= parameter on this JOB statement.

3. Specify the global options for IBM File Export for z/OS by filling in the fields on the panel.

If you plan to use IBM File Export with DB2 databases and you wish to configure DB2 options now, see "Specify DB2 product options" on page 105. If you will not use DB2, or want to configure DB2 options later, you have now finished your product setup activities.

Part 4. The IBM File Export Repository

Chapter 5. Working with the IBM File Export

Repository	35	Create or edit VSAM storage	44
Summary	35	Working with a DB2 storage structure	45
The Repository Management panel	35	Open a DB2 Storage Structure	45
Creating a new repository	36	Working with an IMS storage structure	47
Deleting an existing repository	37	Open an IMS Storage Structure	47
Opening an existing repository	37	Working with VSAM or Sequential records	49
Create a storage structure	38	Working with record layouts	49
Delete a storage structure	39	Working with DB2 record layouts	52
Refresh the storage structure list	39	Working with IMS record layouts	63
Define record layouts	39	Working with VSAM and sequential record layouts	78
Open a storage structure	39	Managing Record Relationships	92
Add or edit a storage structure	39	Specifying DB2 table and RI definitions	94
Create or edit DB2 storage	40	Using the parser	96
Create or edit IMS storage	41		
Create or edit sequential storage	44		

This part describes the IBM File Export Repository and its functions.

Chapter 5. Working with the IBM File Export Repository

IBM File Export's repository permits you to store information about applications and data structures that are used by more than one export. You can store data such as file layouts, record layouts, DB2 tables, and so forth. The use of the repository is optional. However, its use improves your productivity by permitting you to store information once for use by multiple applications.

Summary

Creating a repository

"Creating a new repository" on page 36

Deleting a repository

"Deleting an existing repository" on page 37

Working with the repository

"Opening an existing repository" on page 37

- "Create a storage structure" on page 38
- "Delete a storage structure" on page 39
- "Refresh the storage structure list" on page 39
- "Add or edit a storage structure" on page 39
- "Working with a DB2 storage structure" on page 45
- "Working with an IMS storage structure" on page 47
- "Working with VSAM or Sequential records" on page 49
- "Working with record layouts" on page 49
- "Managing Record Relationships" on page 92
- "Using the parser" on page 96

The Repository Management panel

Selecting **1** on the IBM File Export Main Menu opens the Repository Management panel.

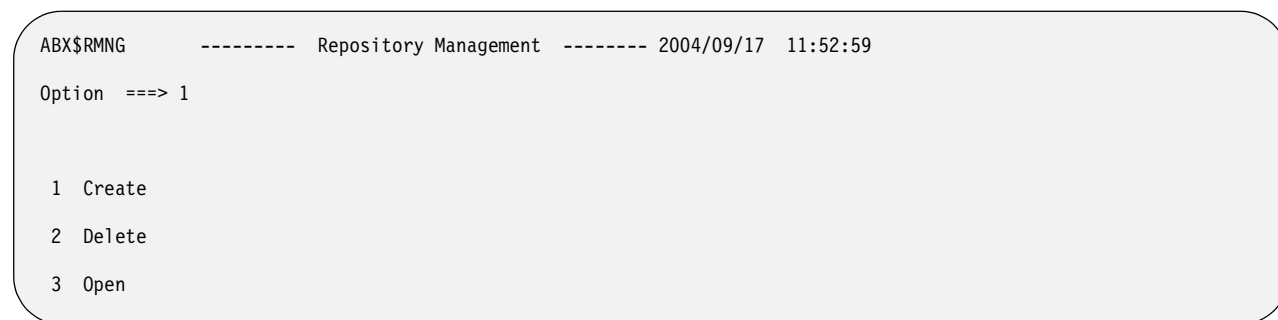


Figure 5. Repository Management (ABX\$RMNG) panel

Panel fields:

- 1 Specifying **1** on the Option line displays the Create Repository (ABX\$ZCRR) panel, from which you can create your IBM File Export repository.

- 2 Specifying **2** on the Option line displays the Delete Repository (ABX\$ZDLR) panel, from which you can delete the repository.
- 3 Specifying **3** on the Option line displays the Repository Selection (ABX\$ZRNM) panel, from which you can open an existing repository for administration.

Creating a new repository

To create a new repository:

1. Select Option **1** on the Repository Management (ABX\$RMNG) panel. The Create Repository (ABX\$ZCRR) panel opens.

```

ABX$ZCRR  V1R1  -----  Create Repository  -----  2004/12/06  11:59:16
Option  ===>

Repository HLQ
Version Number
Repository Name                               >
Number of Record Layouts
Average Number of Fields per Record
Enter "/" to set as default
>

```

Figure 6. Create Repository (ABX\$ZCRR) panel

Panel fields:

Repository HLQ

Specify the high level qualifier to be used in creating this repository.

Version Number

Specify a version number to be assigned to this repository.

Repository Name

Specify a repository name to be assigned to this version of the repository.

Number of Record Layouts

Specify a value to indicate the anticipated number of record layouts that the repository will contain. Valid value must be greater than 0 and less than 999999.

Average Number of Fields per Record

Specify a value for the average number of fields per record that will be stored in the repository. Valid value must be greater than 0 and less than 999999.

Set as Default

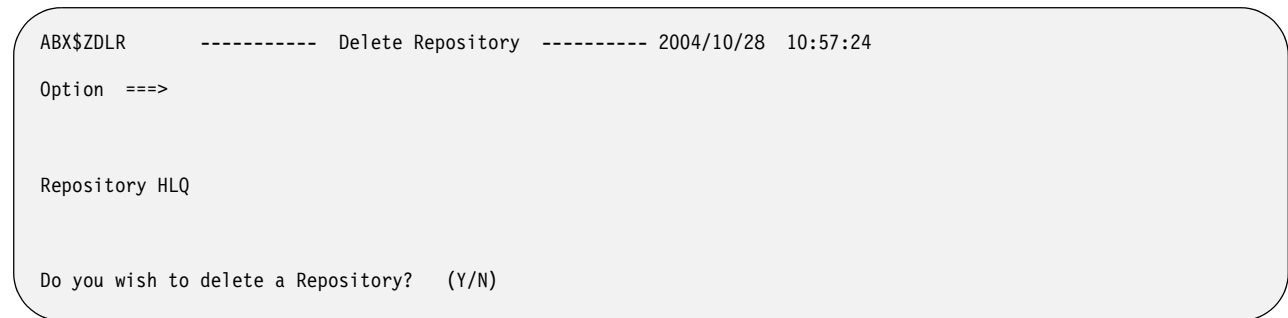
Place a slash (/) on the line to indicate that this is to be used as your default repository.

2. Press **PF3** when your work on this panel is complete. IBM File Export executes a program to create the repository you have specified and returns you to the Repository Management panel.

Deleting an existing repository

To delete an existing repository:

1. Select **Option 2** on the Repository Management (ABX\$RMNG) panel. The Delete Repository (ABX\$ZDLR) panel opens. If you have an existing repository, its name will be displayed in the Repository HLQ field.



```
ABX$ZDLR      ----- Delete Repository ----- 2004/10/28 10:57:24
Option  ===>

Repository HLQ

Do you wish to delete a Repository?  (Y/N)
```

Figure 7. Delete Repository (ABX\$ZDLR) panel

Panel fields:

Repository HLQ

If a high-level qualifier is not already specified in this field, type the high level qualifier to be used to specify the repository to be deleted. If a high-level qualifier is specified, but you want to work with another repository, you can overtype the value in this field.

Do you want to delete a repository?

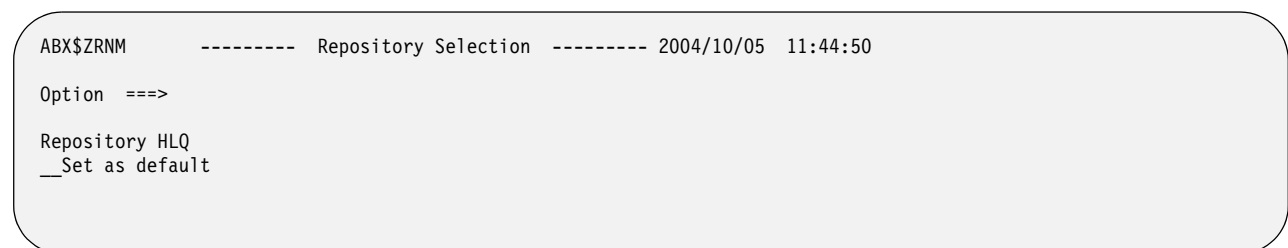
Type **Y** to delete the named repository, or **N** to continue without a deleting a repository.

2. Press **PF3** when your work on this panel is complete. IBM File Export requests that you confirm the deletion and then returns you to the Repository Management panel.

Opening an existing repository

To open an existing repository:

1. Select **Option 3** on the Repository Management (ABX\$RMNG) panel. The Repository Selection (ABX\$ZRNM) panel opens. If you have an existing repository, its name will be displayed in the Repository HLQ field.



```
ABX$ZRNM      ----- Repository Selection ----- 2004/10/05 11:44:50
Option  ===>

Repository HLQ
__Set as default
```

Figure 8. Repository Selection (ABX\$ZRNM) panel

Panel fields:

Repository HLQ

If a high-level qualifier is not already specified in this field, type the high level qualifier to be used to specify the repository to be opened. You may change the value in this field by overtyping it.

Set as default

Type a slash (/) on the line provided to set this repository as your default repository.

2. You can now move through the repository panels. You can create, delete, or edit existing repository structures using the panels provided.

Create a storage structure

If you selected Open Repository, IBM File Export displays the Storage List (ABX\$ZSTL) panel. If you have already created storage for a repository structure, that structure is listed on this panel.

```
ABX$ZSTL      ----- Storage List ----- 2004/10/05 12:10:05
Option  ==>>                               Scroll ==> PAGE

Repository HLQ
Storage Name           >
Storage Type

Row 1 of nn

Cmd  Storage Name           Type
    DB2STORE                DB2ZOS

Valid Line Commands : E - Edit D - Delete O - Open
Valid Option Commands: A - Add R - Refresh L - Record Layouts
```

Figure 9. Storage List (ABX\$ZSTL) panel

Panel Fields:

Repository HLQ

The repository's high-level qualifier is displayed in this field.

Storage Name

This field filters, by name or partial name, the list of storage structures displayed for you on the panel. You will need to type **R** on the Option line to display the filtered list.

Storage Type

This field filters, by storage type, the list of storage structures displayed for you on the panel. You will need to type **R** on the Option line to display the filtered list.

Panel Columns:

Cmd Type any line command available on this panel in this column next to the desired entry.

Storage Name (as displayed in the list)

An arbitrary name assigned to this storage structure.

Storage Type (as displayed in the list)

The storage type, which may be **DB2**, **IMS**, **VSAM**, or **SEQ**.

Valid Line Commands:

- E** Use to edit an entry in this list. When you type the **E** line command in the **Cmd** field, the Edit *mmmm* Storage panel opens, where *mmmm* is the type of structure, for example, IMS or DB2.
- D** Use to delete an entry from this list.
- O** Use to open an entry in this list.

Valid Option Commands:

- A** Use to add a storage structure to the list. When you type the **A** option, the Create Storage (ABX\$ZSTR) panel opens.
- R** Use to refresh the list. If you have added any storage structures, they are now displayed in the list.
- L** Use to go directly to the Record Layouts (ABX\$ZRDL) panel, on which any previously-defined record layouts are listed.

Delete a storage structure

To delete a storage structure from the repository, on the Storage List (ABX\$ZSTL) panel, type **D** on the Cmd line next to the structure you want to delete.

Refresh the storage structure list

To refresh the storage structure list, on the Storage List (ABX\$ZSTL) panel, type **R** on the Option line.

Define record layouts

To define record layouts for a storage structure , on the Storage List (ABX\$ZSTL) panel, type **L** on the Option line. See “Working with record layouts” on page 49.

Open a storage structure

To open a storage structure to work with or modify its substructures, type **O** on the Cmd line of the structure you want to work with. Information can be found in the sections “Open a DB2 Storage Structure” on page 45, “Open an IMS Storage Structure” on page 47, or “Working with VSAM or Sequential records” on page 49.

Add or edit a storage structure

If you want to add a storage structure to the list:

1. Type **A** on the Option line of the Storage List panel.
2. IBM File Export opens the Create Storage (ABX\$ZSTR) panel.

```

ABX$ZSTR      ----- Create Storage ----- 2004/10/28 13:26:27
Option  ==>>

Repository HLQ

Storage Name                                     >

Select storage type

 1 Create DB2 Storage
 2 Create IMS Storage using Databases from a DBDLIB
 3 Create IMS Storage using Databases from an ACBLIB
 4 Create Sequential Storage
 5 Create VSAM Storage

```

Figure 10. Create Storage (ABX\$ZSTR) panel

Panel Fields:

Repository HLQ

The repository's high-level qualifier is displayed in this field.

Storage Name

If you are adding a new storage structure, specify an arbitrary name to be assigned to this structure.

Storage Type

Defines the selections you can make on this panel.

- 1 Specifying 1 on the Option line displays a panel on which you can define the parameters for DB2 storage.
 - 2 Specifying 2 on the Option line displays a panel on which you can define the parameters for IMS storage when the databases reside in a DBDLIB.
 - 3 Specifying 3 on the Option line displays a panel on which you can define the parameters for IMS storage when the databases reside in an ACBLIB.
 - 4 Specifying 4 on the Option line displays a panel on which you can define the parameters for Sequential storage.
 - 5 Specifying 5 on the Option line displays a panel on which you can define the parameters for VSAM storage.
3. Typing a number on the Option line of the Create Storage panel results in displaying the Edit *mmmm* Storage panel for the type of storage you want to create.
 4. If you want to edit a storage structure, typing E on a line of the Storage List panel also opens the Edit *mmmm* Storage panel.

Create or edit DB2 storage

To create new DB2 storage or to edit an existing storage structure:

1. Select Option 1 on the Create Storage (ABX\$ZSTR) panel. You must also enter a name for the storage you are creating in the Storage Name field. To edit an existing storage structure, type E on the Cmd line of an existing storage structure on the Storage List (ABX\$ZSTL) panel.
2. IBM File Export opens the Edit DB2 Storage (ABX\$ZSE1) panel.

```
ABX$ZSE1      ----- Edit DB2 Storage ----- 2004/10/05 13:10:37
Option  =====>

Repository HLQ  PDBURTS.ABX.R1
Storage Name    CSDB2                                     >

DB2 Storage Info:
  DB2 Subsystem
  Plan Name
  DB2 Loadlib
  Additional Loadlib
  Additional Loadlib
```

Figure 11. Edit DB2 Storage (ABX\$ZSE1) panel

Panel Fields:

Repository HLQ
The repository's high-level qualifier is displayed in this field.

Storage Name
The storage name you chose on the Create Storage panel is displayed in this field.

DB2 Subsystem
Type the name of the DB2 subsystem in this field.

Plan Name
Type the name of the DB2 plan in this field.

DB2 Loadlib
Type the name of the DB2 LOADLIB in this field.

Additional Loadlib
Type the names of any optional DB2 LOADLIBs in these two fields.

- 3. Fill out or edit the fields on this panel. When you have finished your work, press PF3 to return to the Storage List panel.

Create or edit IMS storage

To create new IMS storage or to edit an existing storage structure:

- 1. Select Option 2 or Option 3 on the Create Storage (ABX\$ZSTR) panel. You must also enter a name for the storage you are creating in the Storage Name field. To edit an existing storage structure, type E on the Cmd line of an existing storage structure on the Storage List (ABX\$ZSTL) panel.
- 2. If you selected Option 2, Create IMS Storage using Databases from a DBDLIB, IBM File Export opens the Edit IMS Storage (ABX\$ZSE2) panel.



Figure 12. Edit IMS Storage (ABX\$ZSE2) panel

Panel Fields:

Repository HLQ

The repository's high-level qualifier is displayed in this field.

Storage Name

The storage name you chose on the Create Storage panel is displayed in this field.

DBD Loadlib

Type the fully-qualified name of the IMS DBD LOADLIB in this field.

DBD Member

Type the name of the member that will contain this storage definition in this field.

3. If you selected Option 3, Create IMS Storage using Databases from a ACBLIB, IBM File Export opens the Edit IMS Storage (ABX\$ZSE5) panel.


```

ABX$ZSE5  V1R1  -----  Edit IMS Storage  -----  2004/11/09  09:51:58
Option  ==>>

Repository HLQ  PDBURTS.ABX.R1
Storage Name    IMSACB2                                     >

IMS Storage Info:
Do you want to use ACBLIB or MODSTAT?  A  (A or M)

ACBLIB

MODSTAT

IMSACBA

IMSACBB

DATABASE          (Blank for member list)

```

Figure 13. Edit IMS Storage (ABX\$ZSE5) panel

Panel Fields:

Repository HLQ

The repository's high-level qualifier is displayed in this field.

Storage Name

The storage name you chose on the Create Storage panel is displayed in this field.

Do you want to use ACBLIB or MODSTAT

Type **A** to use the ACBLIB or **M** to use MODSTAT.

ACBLIB

Specify the name of the IMS ACBLIB data set in this field. This field is mandatory unless you choose to specify the MODSTAT, ACBLIBA, and ACBLIBB fields. In that case, leave this ACBLIB field blank.

MODSTAT

Specify the name of the IMS MODSTAT data set in this field. Leave this field blank if you fill in the ACBLIB field. If you fill in this field, you must also specify the IMSACBA and IMSACBB fields.

ACBLIBA

Specify the name of the IMS ACBLIBA data set in this field. Leave this field blank if you fill in the ACBLIB field. If you fill in this field, you must also specify the ACBLIBB and MODSTAT fields.

ACBLIBB

Specify the name of the IMS ACBLIBA data set in this field. Leave this field blank if you fill in the ACBLIB field. If you fill in this field, you must also specify the ACBLIBA and MODSTAT fields.

Database

Specify the name of the IMS data base. If you would rather select the target data base from a list, leave this field blank. When you press **Enter**, a panel displays on which you can make a selection from the list of databases presented to you.

4. Fill out or edit the fields on the panel you are presented with. When you have finished your work, press **PF3** to return to the Storage List panel.

Create or edit sequential storage

To create new sequential storage or to edit an existing storage structure:

1. Select Option 4 on the Create Storage (ABX\$ZSTR) panel. You must also enter a name for the storage you are creating in the Storage Name field. To edit an existing storage structure, type **E** on the Cmd line of an existing storage structure on the Storage List (ABX\$ZSTL) panel.
2. IBM File Export opens the Edit VSAM or Sequential Storage (ABX\$ZSE3) panel.

```

ABX$ZSE3      ---- Edit VSAM Or Sequential Storage --- 2004/10/05 13:37:45

Option  ====>

Repository HLQ

Storage Name                                     >

Storage Info:

Data Set Name

```

Figure 14. Edit VSAM or Sequential Storage (ABX\$ZSE3) panel

Panel Fields:

Repository HLQ

The repository's high-level qualifier is displayed in this field.

Storage Name

The storage name you chose on the Create Storage panel is displayed in this field.

Data Set Name

Type the fully-qualified name of the sequential data set in this field.

3. Fill out or edit the fields on this panel. When you have finished your work, press **PF3** to return to the Storage List panel.

Create or edit VSAM storage

To create new VSAM storage or to edit an existing storage structure:

1. Select Option 5 on the Create Storage (ABX\$ZSTR) panel. You must also enter a name for the storage you are creating in the Storage Name field. To edit an existing storage structure, type **E** on the Cmd line of an existing storage structure on the Storage List (ABX\$ZSTL) panel.
2. IBM File Export opens the Edit VSAM or Sequential Storage (ABX\$ZSE3) panel.

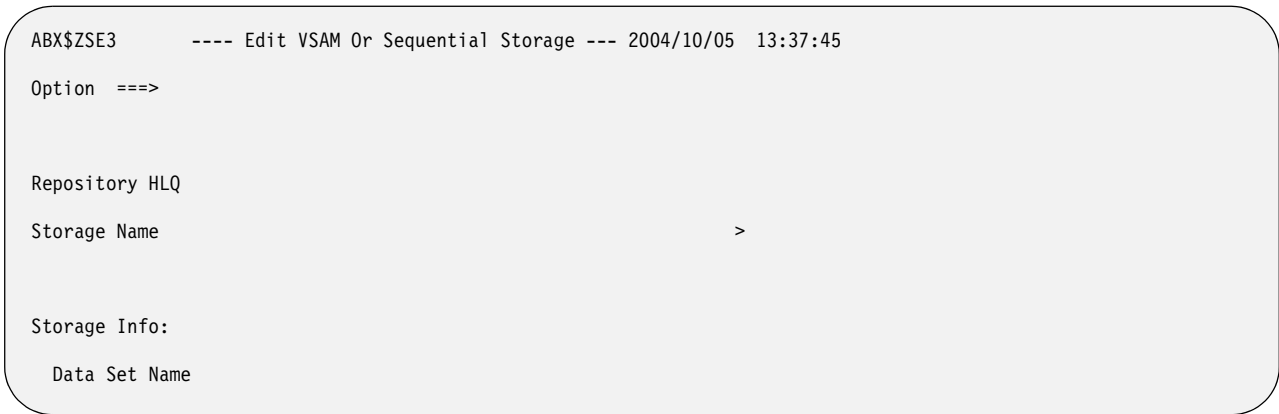


Figure 15. Edit VSAM or Sequential Storage (ABX\$ZSE3) panel

Panel Fields:

Repository HLQ

The repository’s high-level qualifier is displayed in this field.

Storage Name

The storage name you chose on the Create Storage panel is displayed in this field.

Data Set Name

Type the fully-qualified name of the VSAM data set in this field.

3. Fill out or edit the fields on this panel. When you have finished your work, press PF3 to return to the Storage List panel.

Working with a DB2 storage structure

The following section describes how you may work with DB2 when using the repository.

Open a DB2 Storage Structure

To open a storage structure in order to add or edit the components of that structure, on the Storage List (ABX\$ZSTL) panel, type **O** on the Cmd line of the structure you want to work with.

To open a DB2 storage structure:

1. Type **O** on a Cmd line of the Storage List panel for storage whose type is DB2.
2. IBM File Export opens the Table List (ABX\$ZTLS) panel.

```

ABX$ZTLS ----- Table List ----- 2004/10/05 14:04:06
Option ==>                               Scroll ==> PAGE

Repository HLQ
Parent Storage >
Table Name >
Row 1 of n

Cmd Table
ABC.STAFF

Valid Line Commands : D - Delete O - Open
Valid Option Commands: A - Add I - Import R - Refresh

```

Figure 16. Table List (ABX\$ZTLS) panel

Panel Fields:

Repository HLQ

The repository's high-level qualifier is displayed in this field.

Parent Storage

The storage name you chose on the Create Storage panel is displayed in this field.

Table Name

Acts as a filter to limit the display at the bottom of the panel. If you enter a value here, you must type R on the Option line to refresh the display.

Panel Columns:

Cmd Type any line command available on this panel in this column next to the desired entry.

Table This column lists those tables already defined for this storage structure.

Valid Line Commands:

D Use to delete an entry from this list.

O Use to open an entry in this list.

Valid Option Commands:

A Use to add a table to the list.

I Use to specify that you want to import table definitions or RI definitions into your repository.

R Use to refresh the list. If you have added any tables, they are now displayed in the list.

3. To delete a table from the list, type **D** on the Cmd line of the table you want to delete.
4. To refresh the list, type **R** on the Option line.
5. To add a table to the list, type **A** on the Option line of the Table List (ABX\$ZTLS) panel. When you type the **A** option, a pop-up opens on which you can specify the name of the table and the name of the table owner of the table you want to add to the list. If you don't know the name of the table you want, you can enter a partially-qualified table name containing an asterisk. IBM File Export displays the Tables Selection (ABX\$ZTBS) list, from which you can select the table you want to add.

```

ABX$ZTBS      ----- Tables Selection ----- 2004/10/05 14:30:02
Option  ==>>                               Scroll ==> PAGE

Cmd  Owner      Name

Valid Line Commands: S - Select

```

Figure 17. Tables Selection (ABX\$ZTBS) panel

Column Fields:

Cmd Type any line command available on this panel in this column next to the desired entry.

Owner The name of the owner of the table in the Name column.

Name The name of a previously-defined DB2 table.

Valid Line Command:

S Use to select a table to be added to the Table List (ABX\$ZTLS) panel.

6. To open any table on the Table List (ABX\$ZTLS) panel, type **O** on the Cmd line of that table. IBM File Export opens the Record Layouts (ABX\$ZTRL) panel. For information about populating DB2 storage using IBM File Export’s record layout panels, see “Working with record layouts” on page 49.
7. To import table definitions or RI definitions into the repository, type **I** on the Option line of the Table List (ABX\$ZTLS) panel. IBM File Export opens a panel on which you can specify the definitions to be imported. See “Specifying DB2 table and RI definitions” on page 94.

Working with an IMS storage structure

The following section describes how you may work with IMS when using the repository.

Open an IMS Storage Structure

To open an IMS storage structure:

1. Type **O** on a Cmd line of the Storage List panel for storage whose type is IMS.
2. IBM File Export opens the Segment List (ABX\$ZSLS) panel.

```

ABX$ZSLS ----- Segment List ----- 2004/10/05 15:17:25
Option ==>                               Scroll ==> PAGE

Repository HLQ
Parent Storage                             >
Segment Name                               >

Cmd Segment

Valid Line Commands : D - Delete O - Open
Valid Option Commands: A - Add R - Refresh

```

Figure 18. Segment List (ABX\$ZSLS) panel

Panel Fields:

Repository HLQ

The repository's high-level qualifier is displayed in this field.

Parent Storage

The storage name you chose on the Create Storage panel is displayed in this field.

Segment Name

Acts as a filter to limit the display at the bottom of the panel. If you enter a value here, you must type R on the Option line to refresh the display.

Panel Columns:

Cmd Type any line command available on this panel in this column next to the desired entry.

Segment

This column lists those segments already defined for this storage structure.

Valid Line Commands:

D Use to delete an entry from this list.

O Use to open an entry in this list.

Valid Option Commands:

A Use to add a table to the list.

R Use to refresh the list. If you have added any tables, they are now displayed in the list.

3. To delete a segment from the list, type **D** on the Cmd line of the segment you want to delete.
4. To refresh the list, type **R** on the Option line.
5. To add a segment to the list, type **A** on the Option line of the Segment List (ABX\$ZSLS) panel. IBM File Export displays the Segments Selection (ABX\$ZSGS) list, from which you can select the segment you want to add.

```
ABX$ZSGS      ----- Segments Selection ----- 2004/10/05 15:26:16
Option  ==>>                               Scroll ==>> PAGE

Cmd Segment

Valid Line Commands : S - Select
```

Figure 19. Segment Selection (ABX\$ZSGS) panel

Column Fields:

Cmd Type any line command available on this panel in this column next to the desired entry.

Segment The name of a previously-defined IMS segment.

Valid Line Command:

S Use to select a segment to be added to the Segment List (ABX\$ZSLS) panel.

- 6. To open any segment on the Segment List (ABX\$ZSLS) panel, type **O** on the Cmd line of that segment.
- 7. IBM File Export opens the Record Layouts (ABX\$ZFRL) panel. For information about populating IMS storage using IBM File Export’s record layout panels, see “Working with record layouts”.

Working with VSAM or Sequential records

To open a VSAM or sequential storage structure:

- 1. Type **O** on a Cmd line of the Storage List panel for storage whose type is VSAM or sequential.
- 2. IBM File Export opens the Record Layouts (ABX\$ZFRL) panel. See “Working with record layouts”.

Working with record layouts

This section describes how you will work with the Record Layouts panels to further define your repository structures. To see a list of all of the record layouts you have defined for this repository, type **L** on the Options line of the Storage List (ABX\$ZSTL) panel. The Record Layouts panel opens.

```

ABX$ZRDL ----- Record Layouts ----- 2004/10/05 14:26:12
Option ==>                               Scro11 ==> PAGE

Repository HLQ

Record Name                               >

                Suppress Default Layouts

Cmd Record Name

Valid Line Commands : C - Copy E - Edit D - Delete O - Open
Valid Option Commands: N - New P - Parse R - Refresh

```

Figure 20. Record Layouts (ABX\$ZRDL) panel

Panel Fields:

Repository HLQ

The repository's high-level qualifier is displayed in this field.

Record Name

Acts as a filter to limit the display at the bottom of the panel. If you enter a value here, you must type R on the Option line to refresh the display.

Suppress Default Layouts

Place a slash on the line preceding this field if you want to filter the display to omit showing defaults. You will need to refresh the display for the choice to take effect.

Panel Columns:

Cmd Type any line command available on this panel in this column next to the desired entry.

Record Name

This column lists those records already defined for this table.

Valid Line Commands:

- C** Use to copy the specifications that describe an existing record to a new record which you will name.
- E** Use to edit an entry on this list.
- D** Use to delete an entry from this list.
- O** Use to open an entry in this list.

Valid Option Commands:

- N** Use to add a record to the list. The Create Record panel opens.
- P** Use to configure the parser you will use when working with this repository. For information, see "Using the parser" on page 96.

Note: If you select this option, the repository will close until you have completed configuring the parser. Then you will be returned automatically to the panel you were working with.

- R** Use to refresh the list. If you have added any records, they are now displayed in the list.

1. If you typed **C** on the Cmd line, the Copy Record (ABX\$CPR) panel opens. The display fields on this panel are the repository high level qualifier and the name of the source record which you picked on this Record Layouts (ABX\$ZRD) panel. You must supply a name for the record to which the source record specifications will be copied.
2. If you typed **E** on a Cmd line, the Edit Record (ABX\$ZSE4) panel opens for the record you have selected. On this panel, you can rename the record you selected by typing its name on the Record Name line. If a name appears on the line, you can change it by overtyping it. When you have completed work on the Edit Record panel, press **PF3** to continue.

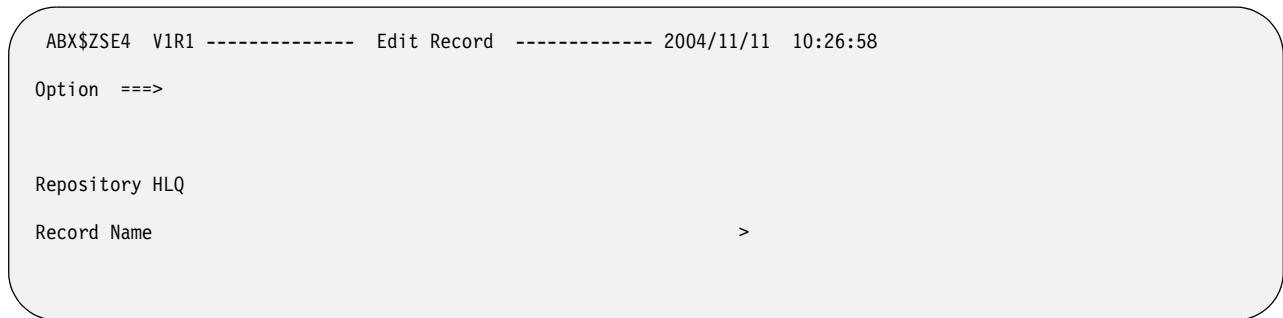


Figure 21. Edit Record (ABX\$ZSE4) panel

Panel Fields:

Repository HLQ

Displays the repository's high-level qualifier.

Record Name

Displays the name of the record to be edited. Change the name by overtyping it.

3. If you typed **O** on a Cmd line, the appropriate record layout panel opens. See the sections that follow that describe the sequence of events for DB2, IMS, sequential, or VSAM records.
4. If you typed **N** on the Option line, the Create Record (ABX\$ZREC) panel opens. On this panel, you can name the new record. When you have completed your work, press **PF3** to continue.

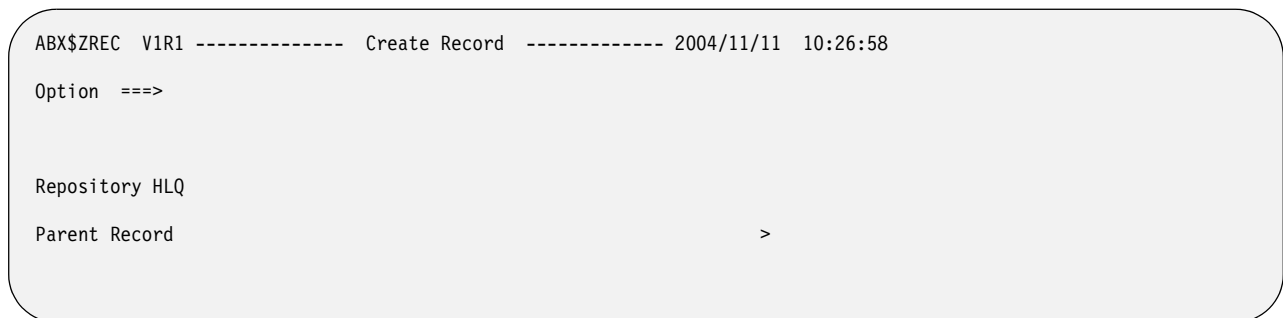


Figure 22. Create Record (ABX\$ZREC) panel

Panel Fields:

Repository HLQ

Displays the repository's high-level qualifier.

Parent Record

Type the name of the parent record.

5. If you typed **P** on the Option line, the Parser Configuration (ABX\$ZPRS) panel opens. See “Using the parser” on page 96.

Working with DB2 record layouts

You can access DB2 record layouts in one of the following ways:

1. Follow these steps from the Options line of the Storage List (ABX\$ZSTL) panel:
 - a. Type **L** on the Options line of the Storage List (ABX\$ZSTL) panel. The Record Layouts (ABX\$ZRDL) panel opens (see page 50).
 - b. Type an **O** next to a DB2 record.
 - c. If this record is a default record, the fields that comprise it cannot be changed. IBM File Export takes you directly to the Column List panel. The Column List (ABX\$ZCNL) panel opens.

```

ABX$ZCNL  VIR1 ----- Column List ----- 2004/11/09 12:40:26
Option  ==>                               Scroll ==> PAGE

Repository HLQ
Parent Storage                               >
Parent File                                 >
Parent Record                               >
                                           Row 1 of 7

Column          Type          Length

Valid Option Commands:  R - Refresh
  
```

Figure 23. Column List (ABX\$ZCNL) panel

Panel Fields:

Repository HLQ

Displays the repository’s high-level qualifier.

Parent Storage

Displays the name for this storage structure that you chose on the Create Storage panel; for example, DB2.

Parent File

Displays the name of the DB2 table to which this record belongs.

Parent Record

Displays the name of the DB2 row for which you are defining columns.

Column

Displays the column name.

Type

Displays the column type, for example, INT, VARCHAR, etc.

Length

Displays the length in bytes of the column.

Valid Option Commands:

- R Use to refresh the list. If you have added any records, they are now displayed in the list.
- d. If this is not a default record, you can add or delete fields, or edit existing fields. In that case, IBM File Export opens the Field List (ABX\$ZFDL) panel.

```
ABX$ZFDL  V1R1 ----- Field List ----- 2004/11/10 10:47:27
Option ====>                               Scroll ====> PAGE

Repository HLQ
Parent Record                               >
                                           Row 1 of 2

Cmd  Field                               Type

Valid Line Commands : E - Edit D - Delete
Valid Option Commands: S - Add Simple Field G - Add Group Field R - Refresh
```

Figure 24. Field List (ABX\$ZFDL) panel

Panel Fields:

Repository HLQ

Displays the repository's high-level qualifier.

Parent Record

Displays the name of the IMS parent record.

Field Displays the names of the fields that have been defined for this record.

Type Displays whether this is a **SIMPLE** or **GROUP** record.

Valid Line Commands:

E Specify to edit an entry in this list.

D Specify to delete an entry from the list.

Valid Option Commands:

S Specify to add a simple field to the list.

G Specify to add a group field to the list.

R Use to refresh the list. If you have added any records, they are now displayed in the list.

- e. If you specify **E** to edit an entry on the list, and if this is a **SIMPLE** field, the Edit Simple Field (ABX\$ZESF) panel opens. On this panel, you can update the entries for any text entry field on the panel. When you have completed your work, press **Enter** to continue.

```

ABX$ZESF V1R1 ----- Edit Simple Field ----- 2004/11/11 10:51:15
Option ==>
                                     More:  +
Parent Record                       >
Field Name

Enter Field information below:
Field Position
Field Length
Field Type

Occurs
Depending On
Precision
Scale
Signed          N (Y/N)
Scramble        N (Y/N)
Group Length
Maximum Length
Position Algorithm
Group Length Algorithm

Enter "/" to select option
Specify additional Position Algorithm
Specify additional Group Length Algorithm

```

Figure 25. Edit Simple Field (ABX\$ZESF) panel

Panel Fields:

Parent Record

Displays the name of the parent record to which this field belongs.

Field Name

Displays the name of the IMS field you want to edit.

Field Position

Specify the position of the field in the record. Position can be specified:

- As an integer type constant like 1 or 100.
- Relative to the end of the record, such as 'end - 10', which indicates ten bytes from the end of the record.
- As a hex constant.

Field Length

Displays the length in bytes of the field.

Type Specify the field's type, such as CHAR or INT.

Occurs

Specify an integer that indicates the number of times a variable length repeating group occurs.

Depending On

Specify the object of the Occurs. If the number of times a repeating group can occur in the record depends on this object of the Occurs, that dependency should be specified here. For information about Occurs Depending On, see "Understanding Occurs Depending On" on page 262.

Precision

Specify the precision of numeric data.

Scale Specify the scale of numeric data.

Signed

Specify whether (Y) or not (N) the numeric data is signed.

Scramble

Specify whether (Y) or not (N) the field should be scrambled.

Group Length

Specify the length of the group.

Maximum Length

Specify the maximum length of the field.

Position Algorithm

Because the length of a group item containing variable length data is not a constant, an algorithm is required to express the position of a field, and in some cases, the length of the field or group item. Specify the position algorithm in this field.

Group Length Algorithm

Because the length of a group item containing variable length data is not a constant, an algorithm is required to express the position of a field, and in some cases, the length of the field or group item. Specify the position algorithm in this field.

Enter "/" to select option

To add or delete a position algorithm, group length algorithm, or both, type a slash (/) in the appropriate field or fields. The Algorithm List (ABX\$ZALG) panel opens.

- f. If you specify **E** to edit an entry on the list, and if this is a GROUP field, the Edit Group Field (ABX\$ZEGF) panel opens. On this panel, you can update the entries for any field on the panel.

ABX\$ZEGF V1R1 ----- Edit Group Field ----- 2004/11/11 10:57:53

Option ==>

More: +

Parent Record >

Field Name

Subrecord >

Enter Field information below:

Field Position

Field Length

Field Type (REDEFINE/SUBREC)

Occurs

Depending On

Scramble N (Y/N)

Group Length

Maximum Length

Position Algorithm

Group Length Algorithm

Enter "/" to select option

Specify additional Position Algorithm

Specify additional Group Length Algorithm

Valid Option Commands: E - Edit Subrecord O - Open Subrecord

Figure 26. Edit Group Field (ABX\$ZEGF) panel

Panel Fields:

Parent Record

Displays the name of the parent record to which this field belongs.

Field Name

Displays the name of the IMS field to which this subfield belongs.

Subrecord

Displays the name of the IMS subrecord you want to edit.

Field Position

Specify the position of the field in the record. Position can be specified:

- As an integer type constant like 1 or 100.

- Relative to the end of the record, such as 'end - 10', which indicates ten bytes from the end of the record.
- As a hex constant.

Field Length

Displays the length in bytes of the field.

Field Type

Specify **REDEFINE** or **SUBREC**. **REDEFINE** specifies that you are defining more than one linked record. **SUBREC** specifies that you are defining a single record.

Occurs

Specify an integer that indicates the number of times a variable length repeating group occurs.

Depending On

Specify the object of the Occurs. If the number of times a repeating group can occur in the record depends on this object of the Occurs, that dependency should be specified here. For information about Occurs Depending On, see "Understanding Occurs Depending On" on page 262.

Scramble

Specify whether (Y) or not (N) the field should be scrambled.

Group Length

Specify the length of the group.

Maximum Length

Specify the maximum length of the field.

Position Algorithm

Because the length of a group item containing variable length data is not a constant, an algorithm is required to express the position of a field, and in some cases, the length of the field or group item. Specify the position algorithm in this field.

Group Length Algorithm

Because the length of a group item containing variable length data is not a constant, an algorithm is required to express the position of a field, and in some cases, the length of the field or group item. Specify the position algorithm in this field.

Enter "/" to select option

To add or delete a position algorithm, group length algorithm, or both, type a slash (/) in the appropriate field or fields. The Algorithm List (ABX\$ZALG) panel opens.

Valid Line Commands:

- E** Use to edit the subrecord.
- O** Use to open a list of fields that comprise this subrecord.

From the Edit Group Field panel, you can choose to:

- Edit the Subrecord by typing **E** on the Option line. The Record Layouts Selection panel opens. Choose the subrecord you want to edit by typing an **S** on its Cmd line. This opens the appropriate Edit *xxxxx* Field panel (either ABX\$ZESF or ABX\$ZEGF), where *xxxxx* is either *SIMPLE* or *GROUP*, where you can perform your edits.
- Open the subrecord by typing **O** on the Option line. The Field List (ABX\$ZFDL) panel opens (see Figure 32 on page 65) so that you can continue your record layout definition.

When you have completed your work, press **Enter** to continue.

- g. If you specify **S** on the Option line to add a SIMPLE field, the Create Simple Field (ABX\$ZESF) panel opens. When you have completed your work, press **Enter** to continue.

```

ABX$ZCSF  V1R1 ----- Create Simple Field ----- 2004/11/11  10:51:15
Option  ==>
                                           More:   +
Parent Record                               >
Field Name
Enter Field information below:
Field Position
Field Length
Field Type
Occurs
Depending On
Precision
Scale
Signed          N  (Y/N)
Scramble        N  (Y/N)
Group Length
Maximum Length
Position Algorithm
Group Length Algorithm
Enter "/" to select option
Specify additional Position Algorithm
Specify additional Group Length Algorithm

```

Figure 27. Create Simple Field (ABX\$ZCSF) panel

Panel Fields:

Parent Record

Displays the name of the parent record to which this field belongs.

Field Name

Displays the name of the IMS field you want to edit.

Field Position

Specify the position of the field in the record. Position can be specified:

- As an integer type constant like 1 or 100.

- Relative to the end of the record, such as 'end - 10', which indicates ten bytes from the end of the record.
- As a hex constant.

Field Length

Displays the length in bytes of the field.

Type Specify the field's type, such as CHAR or INT.

Occurs

Specify an integer that indicates the number of times a variable length repeating group occurs.

Depending On

Specify the object of the Occurs. If the number of times a repeating group can occur in the record depends on this object of the Occurs, that dependency should be specified here. For information about Occurs Depending On, see "Understanding Occurs Depending On" on page 262.

Precision

Specify the precision of numeric data.

Scale Specify the scale of numeric data.

Signed

Specify whether (Y) or not (N) the numeric data is signed.

Scramble

Specify whether (Y) or not (N) the field should be scrambled.

Group Length

Specify the length of the group.

Maximum Length

Specify the maximum length of the field.

Position Algorithm

Because the length of a group item containing variable length data is not a constant, an algorithm is required to express the position of a field, and in some cases, the length of the field or group item. Specify the position algorithm in this field.

Group Length Algorithm

Because the length of a group item containing variable length data is not a constant, an algorithm is required to express the position of a field, and in some cases, the length of the field or group item. Specify the position algorithm in this field.

Enter "/" to select option

To add or delete a position algorithm, group length algorithm, or both, type a slash (/) in the appropriate field or fields. The Algorithm List (ABX\$ZALG) panel opens.

- h. If you specify **G** on the Option line to add a GROUP field, the Create Group Field (ABX\$ZEGF) panel opens. When you have completed your work, press **Enter** to continue.

ABX\$ZCGF V1R1 ----- Create Group Field ----- 2004/11/11 10:57:53

Option ==>

More: +

Parent Record >

Field Name >

Subrecord >

Enter Field information below:

Field Position

Field Length

Field Type (REDEFINE/SUBREC)

Occurs

Depending On

Scramble N (Y/N)

Group Length

Maximum Length

Position Algorithm

Group Length Algorithm

Enter "/" to select option

Specify additional Position Algorithm

Specify additional Group Length Algorithm

Figure 28. Create Group Field (ABX\$ZCGF) panel

Panel Fields:

Parent Record

Displays the name of the parent record to which this field belongs.

Field Name

Displays the name of the IMS field to which this subfield belongs.

Field Name

Type the name of the IMS subrecord you want to create.

Field Position

Specify the position of the field in the record. Position can be specified:

- As an integer type constant like 1 or 100.

- Relative to the end of the record, such as 'end - 10', which indicates ten bytes from the end of the record.
- As a hex constant.

Field Length

Displays the length in bytes of the field.

Field Type

Specify **REDEFINE** or **SUBREC**. **REDEFINE** specifies that you are defining more than one linked record. **SUBREC** specifies that you are defining a single record.

Occurs

Specify an integer that indicates the number of times a variable length repeating group occurs.

Depending On

Specify the object of the Occurs. If the number of times a repeating group can occur in the record depends on this object of the Occurs, that dependency should be specified here. For information about Occurs Depending On, see "Understanding Occurs Depending On" on page 262.

Scramble

Specify whether (Y) or not (N) the field should be scrambled.

Group Length

Specify the length of the group.

Maximum Length

Specify the maximum length of the field.

Position Algorithm

Because the length of a group item containing variable length data is not a constant, an algorithm is required to express the position of a field, and in some cases, the length of the field or group item. Specify the position algorithm in this field.

Group Length Algorithm

Because the length of a group item containing variable length data is not a constant, an algorithm is required to express the position of a field, and in some cases, the length of the field or group item. Specify the position algorithm in this field.

Enter "/" to select option

To add or delete a position algorithm, group length algorithm, or both, type a slash (/) in the appropriate field or fields. The Algorithm List (ABX\$ZALG) panel opens.

2. Follow these steps from a Cmd line of the Storage List (ABX\$ZSTL) panel:
 - a. Type an **O** on the Cmd line of a DB2 Storage Structure. The Table List panel opens. (For more information about the Table List panel, see page 46).

```

ABX$ZTLS      ----- Table List ----- 2004/10/05 14:04:06
Option  ==>                               Scroll ==> PAGE

Repository HLQ
Parent Storage          >
Table Name              >
                               Row 1 of n

Cmd Table

Valid Line Commands : D - Delete  O - Open
Valid Option Commands: A - Add I - Import  R - Refresh

```

Figure 29. Table List (ABX\$ZTLS) panel

- b. Type an **O** on the Cmd line next to a table name on the Table List (ABX\$ZTLS) panel. The Record Layouts (ABX\$ZTRL) panel specific to DB2 opens.

```

ABX$ZTRL V1R1 ----- Record Layouts ----- 2004/11/09 12:54:19
Option  ==>                               Scroll ==> PAGE

Repository HLQ
Parent Storage          >
Parent File            >
Record Name            >
                Suppress Default Layouts
                               Row 1 of 1

Cmd Record Name

Valid Line Commands : O - Open
Valid Option Commands: R - Refresh

```

Figure 30. Record Layouts (ABX\$ZTRL) panel

Panel Fields:

Repository HLQ

Displays the repository's high-level qualifier.

Parent Storage

Displays the name for this storage structure that you chose on the Create Storage panel; for example, DB2.

Parent File

Displays the name of the DB2 table to which this record belongs.

Record Name

Type the name of the record you want to define.

Suppress Default Layouts

Place a slash on the line preceding this field if you want to filter the display to omit showing defaults. You will need to refresh the display for the choice to take effect.

Record Name

Displays the names of any records that have already been defined.

Valid Line Commands:

O Use to open the record next to which it is typed.

Valid Option Commands:

R Use to refresh the list. If you have added any records, they are now displayed in the list.

- c. Type an **O** on the Cmd line next to a record. If this record is a default record, the fields that comprise it cannot be changed. IBM File Export takes you directly to the Column List panel. The Column List (ABX\$ZCNL) panel opens (see Figure 23 on page 52). If this is not a default record, you can add or delete fields, or edit existing fields. In that case, IBM File Export opens the Field List (ABX\$ZFDL) panel (see Figure 23 on page 52).
3. When you have completed your work, pressing **PF3** multiple times will take you back through the panels you have used. You may continue to work with other DB2 storage structures or continue to press **PF3** to exit the repository.

Working with IMS record layouts

You can access IMS record layouts in these ways:

1. Follow these steps from the Options line of the Storage List (ABX\$ZSTL) panel:
 - a. Type **L** on the Options line of the Storage List (ABX\$ZSTL) panel. The Record Layouts (ABX\$ZRDL) panel opens (see page 50).
 - b. Type an **O** next to an IMS record.
 - c. If this record is a default record, the fields that comprise it cannot be changed. IBM File Export takes you directly to the Field List panel. The Field List (ABX\$ZFDD) panel opens.

```

ABX$ZFDD V1R1 ----- Field List ----- 2004/11/10 11:55:03
Option ==>                               Scroll ==> PAGE

Repository HLQ

Parent Storage
Parent file                               >

Parent Record                             >

                                           Row 1 of 3

Field   Start  Size  Type

Valid Option Commands: R - Refresh

```

Figure 31. Field List (ABX\$ZFDD) panel

Panel Fields:

Repository HLQ

Displays the repository's high-level qualifier.

Parent Storage

Displays the name for this storage structure that you chose on the Create Storage panel; for example, IMS01.

Parent File

Displays the name of the IMS segment to which this record belongs.

Record Name

Displays the name of the IMS record.

Field Displays the names of the fields that have been defined for this record.

Start Displays the starting position of the field.

Size Displays the size in bytes of the field.

Type Use to specify the field's type, such as CHAR or INT.

Valid Line Commands:

O Use to open this record to its next level.

Valid Option Commands:

R Use to refresh the list. If you have added any records, they are now displayed in the list.

- d. If this is not a default record, you can add or delete fields, or edit existing fields. In that case, IBM File Export opens the Field List (ABX\$ZFDD) panel.

```

ABX$ZFDL  V1R1 ----- Field List ----- 2004/11/10 10:47:27
Option ==>>                               Scroll ==>> PAGE

Repository HLQ
Parent Record                               >
                                           Row 1 of 2

Cmd  Field                               Type

Valid Line Commands : E - Edit D - Delete
Valid Option Commands: S - Add Simple Field G - Add Group Field R - Refresh

```

Figure 32. Field List (ABX\$ZFDL) panel

Panel Fields:

Repository HLQ
 Displays the repository’s high-level qualifier.

Parent Record
 Displays the name of the IMS parent record.

Field Displays the names of the fields that have been defined for this record.

Type Displays whether this is a **SIMPLE** or **GROUP** record.

Valid Line Commands:

- E** Specify to edit an entry in this list.
- D** Specify to delete an entry from the list.

Valid Option Commands:

- S** Specify to add a simple field to the list.
- G** Specify to add a group field to the list.
- R** Use to refresh the list. If you have added any records, they are now displayed in the list.

e. If you specify **E** to edit an entry on the list, and if this is a **SIMPLE** field, the Edit Simple Field (ABX\$ZESF) panel opens. On this panel, you can update the entries for any text entry field on the panel. When you have completed your work, press **Enter** to continue.

```

ABX$ZESF V1R1 ----- Edit Simple Field ----- 2004/11/11 10:51:15
Option ==>
                                     More:  +
Parent Record                       >
Field Name

Enter Field information below:
Field Position
Field Length
Field Type

Occurs
Depending On
Precision
Scale
Signed          N (Y/N)
Scramble        N (Y/N)
Group Length
Maximum Length
Position Algorithm
Group Length Algorithm

Enter "/" to select option
Specify additional Position Algorithm
Specify additional Group Length Algorithm

```

Figure 33. Edit Simple Field (ABX\$ZESF) panel

Panel Fields:

Parent Record

Displays the name of the parent record to which this field belongs.

Field Name

Displays the name of the IMS field you want to edit.

Field Position

Specify the position of the field in the record. Position can be specified:

- As an integer type constant like 1 or 100.
- Relative to the end of the record, such as 'end - 10', which indicates ten bytes from the end of the record.
- As a hex constant.

Field Length

Displays the length in bytes of the field.

Type Specify the field's type, such as CHAR or INT.

Occurs

Specify an integer that indicates the number of times a variable length repeating group occurs.

Depending On

Specify the object of the Occurs. If the number of times a repeating group can occur in the record depends on this object of the Occurs, that dependency should be specified here. For information about Occurs Depending On, see "Understanding Occurs Depending On" on page 262.

Precision

Specify the precision of numeric data.

Scale Specify the scale of numeric data.

Signed

Specify whether (Y) or not (N) the numeric data is signed.

Scramble

Specify whether (Y) or not (N) the field should be scrambled.

Group Length

Specify the length of the group.

Maximum Length

Specify the maximum length of the field.

Position Algorithm

Because the length of a group item containing variable length data is not a constant, an algorithm is required to express the position of a field, and in some cases, the length of the field or group item. Specify the position algorithm in this field.

Group Length Algorithm

Because the length of a group item containing variable length data is not a constant, an algorithm is required to express the position of a field, and in some cases, the length of the field or group item. Specify the position algorithm in this field.

Enter "/" to select option

To add or delete a position algorithm, group length algorithm, or both, type a slash (/) in the appropriate field or fields. The Algorithm List (ABX\$ZALG) panel opens.

- f. If you specify **E** to edit an entry on the list, and if this is a GROUP field, the Edit Group Field (ABX\$ZEGF) panel opens. On this panel, you can update the entries for any field on the panel.

ABX\$ZEGF V1R1 ----- Edit Group Field ----- 2004/11/11 10:57:53

Option ==>

More: +

Parent Record >

Field Name

Subrecord >

Enter Field information below:

Field Position

Field Length

Field Type (REDEFINE/SUBREC)

Occurs

Depending On

Scramble N (Y/N)

Group Length

Maximum Length

Position Algorithm

Group Length Algorithm

Enter "/" to select option

Specify additional Position Algorithm

Specify additional Group Length Algorithm

Valid Option Commands: E - Edit Subrecord O - Open Subrecord

Figure 34. Edit Group Field (ABX\$ZEGF) panel

Panel Fields:

Parent Record

Displays the name of the parent record to which this field belongs.

Field Name

Displays the name of the IMS field to which this subfield belongs.

Field Name

Displays the name of the IMS subrecord you want to edit.

Field Position

Specify the position of the field in the record. Position can be specified:

- As an integer type constant like 1 or 100.

- Relative to the end of the record, such as 'end - 10', which indicates ten bytes from the end of the record.
- As a hex constant.

Field Length

Displays the length in bytes of the field.

Field Type

Specify **REDEFINE** or **SUBREC**. **REDEFINE** specifies that you are defining more than one linked record. **SUBREC** specifies that you are defining a single record.

Occurs

Specify an integer that indicates the number of times a variable length repeating group occurs.

Depending On

Specify the object of the Occurs. If the number of times a repeating group can occur in the record depends on this object of the Occurs, that dependency should be specified here. For information about Occurs Depending On, see "Understanding Occurs Depending On" on page 262.

Scramble

Specify whether (Y) or not (N) the field should be scrambled.

Group Length

Specify the length of the group.

Maximum Length

Specify the maximum length of the field.

Position Algorithm

Because the length of a group item containing variable length data is not a constant, an algorithm is required to express the position of a field, and in some cases, the length of the field or group item. Specify the position algorithm in this field.

Group Length Algorithm

Because the length of a group item containing variable length data is not a constant, an algorithm is required to express the position of a field, and in some cases, the length of the field or group item. Specify the position algorithm in this field.

Enter "/" to select option

To add or delete a position algorithm, group length algorithm, or both, type a slash (/) in the appropriate field or fields. The Algorithm List (ABX\$ZALG) panel opens.

Valid Line Commands:

- E** Use to edit the subrecord.
- O** Use to open a list of fields that comprise this subrecord.

From the Edit Group Field panel, you can choose to:

- Edit the Subrecord by typing **E** on the Option line. The Record Layouts Selection panel opens. Choose the subrecord you want to edit by typing an **S** on its Cmd line. This opens the appropriate Edit *xxxxx* Field panel (either ABX\$ZESF or ABX\$ZEGF), where *xxxxx* is either *SIMPLE* or *GROUP*, where you can perform your edits.
- Open the subrecord by typing **O** on the Option line. The Field List (ABX\$ZFDL) panel opens (see Figure 32 on page 65) so that you can continue your record layout definition.

When you have completed your work, press **Enter** to continue.

- g. If you specify **S** on the Option line to add a SIMPLE field, the Create Simple Field (ABX\$ZESF) panel opens. When you have completed your work, press **Enter** to continue.

```

ABX$ZCSF V1R1 ----- Create Simple Field ----- 2004/11/11 10:51:15
Option ==>
                                     More:  +
Parent Record                       >
Field Name

Enter Field information below:
Field Position
Field Length
Field Type

Occurs
Depending On
Precision
Scale
Signed          N (Y/N)
Scramble        N (Y/N)
Group Length
Maximum Length
Position Algorithm
Group Length Algorithm

Enter "/" to select option
Specify additional Position Algorithm
Specify additional Group Length Algorithm

```

Figure 35. Create Simple Field (ABX\$ZCSF) panel

Panel Fields:

Parent Record

Displays the name of the parent record to which this field belongs.

Field Name

Displays the name of the IMS field you want to edit.

Field Position

Specify the position of the field in the record. Position can be specified:

- As an integer type constant like 1 or 100.

- Relative to the end of the record, such as 'end - 10', which indicates ten bytes from the end of the record.
- As a hex constant.

Field Length

Displays the length in bytes of the field.

Type Specify the field's type, such as CHAR or INT.

Occurs

Specify an integer that indicates the number of times a variable length repeating group occurs.

Depending On

Specify the object of the Occurs. If the number of times a repeating group can occur in the record depends on this object of the Occurs, that dependency should be specified here. For information about Occurs Depending On, see "Understanding Occurs Depending On" on page 262.

Precision

Specify the precision of numeric data.

Scale Specify the scale of numeric data.

Signed

Specify whether (Y) or not (N) the numeric data is signed.

Scramble

Specify whether (Y) or not (N) the field should be scrambled.

Group Length

Specify the length of the group.

Maximum Length

Specify the maximum length of the field.

Position Algorithm

Because the length of a group item containing variable length data is not a constant, an algorithm is required to express the position of a field, and in some cases, the length of the field or group item. Specify the position algorithm in this field.

Group Length Algorithm

Because the length of a group item containing variable length data is not a constant, an algorithm is required to express the position of a field, and in some cases, the length of the field or group item. Specify the position algorithm in this field.

Enter "/" to select option

To add or delete a position algorithm, group length algorithm, or both, type a slash (/) in the appropriate field or fields. The Algorithm List (ABX\$ZALG) panel opens.

- h. If you specify **G** on the Option line to add a GROUP field, the Create Group Field (ABX\$ZEGF) panel opens. When you have completed your work, press **Enter** to continue.

ABX\$ZCGF V1R1 ----- Create Group Field ----- 2004/11/11 10:57:53

Option ==>

More: +

Parent Record >

Field Name >

Subrecord >

Enter Field information below:

Field Position

Field Length

Field Type (REDEFINE/SUBREC)

Occurs

Depending On

Scramble N (Y/N)

Group Length

Maximum Length

Position Algorithm

Group Length Algorithm

Enter "/" to select option

Specify additional Position Algorithm

Specify additional Group Length Algorithm

Figure 36. Create Group Field (ABX\$ZCGF) panel

Panel Fields:

Parent Record

Displays the name of the parent record to which this field belongs.

Field Name

Displays the name of the IMS field to which this subfield belongs.

Field Name

Type the name of the IMS subrecord you want to create.

Field Position

Specify the position of the field in the record. Position can be specified:

- As an integer type constant like 1 or 100.

- Relative to the end of the record, such as 'end - 10', which indicates ten bytes from the end of the record.
- As a hex constant.

Field Length

Displays the length in bytes of the field.

Field Type

Specify **REDEFINE** or **SUBREC**. **REDEFINE** specifies that you are defining more than one linked record. **SUBREC** specifies that you are defining a single record.

Occurs

Specify an integer that indicates the number of times a variable length repeating group occurs.

Depending On

Specify the object of the Occurs. If the number of times a repeating group can occur in the record depends on this object of the Occurs, that dependency should be specified here. For information about Occurs Depending On, see "Understanding Occurs Depending On" on page 262.

Scramble

Specify whether (Y) or not (N) the field should be scrambled.

Group Length

Specify the length of the group.

Maximum Length

Specify the maximum length of the field.

Position Algorithm

Because the length of a group item containing variable length data is not a constant, an algorithm is required to express the position of a field, and in some cases, the length of the field or group item. Specify the position algorithm in this field.

Group Length Algorithm

Because the length of a group item containing variable length data is not a constant, an algorithm is required to express the position of a field, and in some cases, the length of the field or group item. Specify the position algorithm in this field.

Enter "/" to select option

To add or delete a position algorithm, group length algorithm, or both, type a slash (/) in the appropriate field or fields. The Algorithm List (ABX\$ZALG) panel opens.

2. Follow these steps from a Cmd line of the Storage List (ABX\$ZSTL) panel:
 - a. Type an **O** on the Cmd line of an IMS Storage Structure. The Segment List panel opens. (For more information about the Segment List panel, see page 48).

```

ABX$ZSLS V1R1 ----- Segment List ----- 2004/11/10 11:59:08
Option ==>                               Scroll ==> PAGE

Repository HLQ
Parent Storage                             >
Segment Name                               >
                                           Row 1 of 4

Cmd Segment

Valid Line Commands : D - Delete O - Open
Valid Option Commands: A - Add R - Refresh

```

Figure 37. Segment List (ABX\$ZSLS) panel

- b. Type **O** on the Cmd line next to a segment name on the Segment List (ABX\$ZSLS) panel. The Record Layouts (ABX\$ZFRL) panel specific to IMS opens.

```

ABX$ZFRL V1R1 ----- Record Layouts ----- 2004/11/10 10:25:14
Option ==>                               Scroll ==> PAGE

Repository
Parent Storage                             >
Parent File                               >
Record Name                               >
                                           Row 1 of 1

Suppress Default Layouts

Cmd Record Name

Valid Line Commands : U - Unlink M - Manage O - Open
Valid Option Commands: N - New R - Refresh L - Link

```

Figure 38. Record Layouts (ABX\$ZFRL) panel

Panel Fields:

Repository HLQ

Displays the repository's high-level qualifier.

Parent Storage

Displays the name for this storage structure that you chose on the Create Storage panel; for example, ABXIMS.

Parent File

Displays the name of the IMS file to which this record belongs.

Record Name

Acts as a filter to limit the display at the bottom of the panel. If you enter a value here, you must type R on the Option line to refresh the display.

Suppress Default Layouts

Place a slash on the line preceding this field if you want to filter the display to omit showing defaults. You will need to refresh the display for the choice to take effect.

Record Name

Displays the names of any records that have already been defined.

Valid Line Commands:

- U** Use to unlink (remove) a record from the list.
- M** Use to add selection criteria or define relationship specifications for the selected record.
- O** Use to open the record to lower-level subrecord.

Valid Option Commands:

- N** Use to add a record to the list. You will also need to specify a name for the record in the field provided at the top of the panel.
 - R** Use to refresh the list. If you have added any records, they are now displayed in the list.
 - L** Use to link (add) another record to the list. A list of records available to be linked will be displayed.
- c. Type **U** on the Cmd line next to a record to unlink (remove) that record from the list of records comprising the segment.
- d. Type **M** on the Cmd line next to a record to add or delete selection criteria for that record or to define relationship specifications for the record. The Manage Record Layouts panel opens.

```

ABX$ZMRL V1R1 ----- Manage Record Layouts ----- 2004/12/22 11:54:13

Option ==>

Repository HLQ
Parent Storage >
Parent File >
Current Record >

Data specification
  1 Edit Selection Criteria
  2 Relationship Specification

```

Figure 39. Manage Record Layouts (ABX\$ZMRL) panel

Panel Fields:

Repository HLQ

Displays the repository's high-level qualifier.

Parent Storage

Displays the name for this storage structure that you chose on the Create Storage panel; for example, ABXVSM1.

Parent Storage

Displays the name of the parent storage structure to which this record belongs.

Parent File

Displays the name of the parent file to which this record belongs.

Current Record

Displays the name of the current record.

Valid Line Commands:

- 1 Displays a panel on which you can define selection criteria
- 2 Displays a panel on which you can begin to define record relationships.

On this panel, selecting Option 1 takes you to the Selection Criteria (ABX\$ZSCR) panel. Selecting Option 2 takes you to the Manage Record Relations (ABX\$ZREL) panel. To understand the panel flow to define record relationships, see "Managing Record Relationships" on page 92.

- e. When you select Option 1, the Selection Criteria (ABX\$ZSCR) panel opens. On this panel you can add or delete selection criteria for the record. A selection criterion typically is entered as a WHERE clause. Note that, when specifying a value to be matched in the WHERE clause, that value is case-sensitive. If you enter state='Alabama', IBM File Export will not select records where state='alabama'.

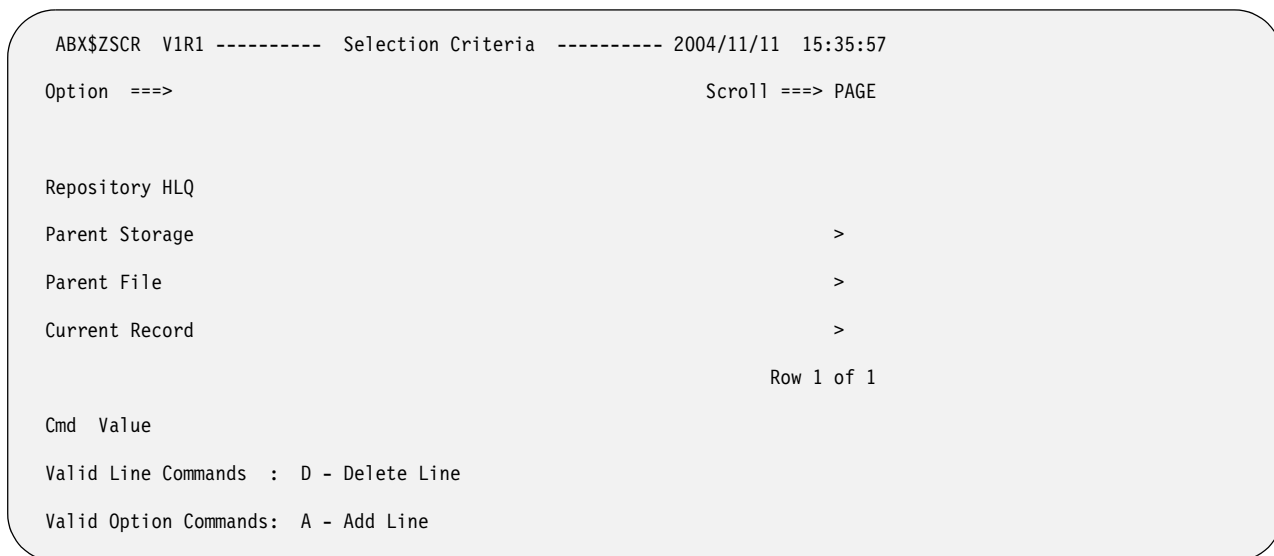


Figure 40. Selection Criteria (ABX\$ZSCR) panel

Panel Fields:

Repository HLQ

Displays the repository's high-level qualifier.

Parent Storage

Displays the name for this storage structure that you chose on the Create Storage panel; for example, ABXVSM1.

Parent File

Displays the name of the file to which this record belongs.

Record Name

Displays the name of the current record.

Cmd Enter any valid line command in this field.

Value

Type the value to be matched as the selection criteria, typically in the form of a WHERE clause.

Valid Line Commands:

D Deletes the selected value from the list.

Valid Option Commands:

A Adds a blank line on which you can enter a value to be matched.

- f. Type **O** on the Cmd line next to a record to open that record's list of fields. If this record is a default record, the fields that comprise it cannot be changed. IBM File Export takes you directly to the Field List panel. The Field List (ABX\$ZFDD) panel opens (see Figure 31 on page 64). If this is not a default record, you can add or delete fields, or edit existing fields. In that case, IBM File Export opens the Field List (ABX\$ZFDL) panel (see Figure 32 on page 65).
- g. Type **N** on the Option line to add a record. The Create Record panel (ABX\$ZREC, see Figure 22 on page 51) opens.
- h. Type **R** on the Option line to refresh the panel. The panel will be updated with any additions or deletions since the last refresh.
- i. Type **L** on the Option line to link additional records to the segment. The Record Layouts Selection (ABX\$ZRDS) panel opens. Choose the record you want to link by typing **S** on the Cmd line next to its name.



Figure 41. Record Layouts Selection (ABX\$ZRDS) panel

Panel Fields:**Repository HLQ**

Displays the repository's high-level qualifier.

Record Name

Acts as a filter to limit the display at the bottom of the panel. If you enter a value here, you must type R on the Option line to refresh the display.

Cmd Enter any valid line command in this field.

Suppress Default Layouts

Place a slash on the line preceding this field if you want to filter the display to omit showing defaults. You will need to refresh the display for the choice to take effect.

Record Name

Displays a list of records from which you can choose records to be added to the segment.

Valid Line Commands:

S Selects the record to be added.

Valid Option Commands:

R Refreshes the list to display any added record layouts.

3. When you have completed your work, pressing **PF3** multiple times will take you back to the Storage List panel. You may continue to work with other IMS storage structures or press **PF3** to exit the repository.

Working with VSAM and sequential record layouts

You can access sequential or VSAM record layouts in these ways.

1. Follow these steps from the Options line of the Storage List (ABX\$ZSTL) panel:
 - a. Type **L** on the Options line of the Storage List (ABX\$ZSTL) panel. The Record Layouts (ABX\$ZRDL) panel opens (see page 50).
 - b. Type an **O** next to a sequential or VSAM record.
 - c. If this record is a default record, the fields that comprise it cannot be changed. IBM File Export takes you directly to the Field List panel. The Field List (ABX\$ZFDD) panel opens.

```

ABX$ZFDD V1R1 ----- Field List ----- 2004/11/10 11:55:03
Option ==>                               Scroll ==> PAGE

Repository HLQ
Parent Storage
Parent file                               >
Parent Record                             >
                                           Row 1 of 3

Field   Start  Size  Type
Valid Option Commands: R - Refresh
  
```

Figure 42. Field List (ABX\$ZFDD) panel

Panel Fields:

Repository HLQ

Displays the repository's high-level qualifier.

Parent Storage

Displays the name for this storage structure that you chose on the Create Storage panel; for example, SEQ01.

Parent File

Displays the name of the file to which this record belongs.

Record Name

Displays the name of the record.

Field Displays the names of the fields that have been defined for this record.

Start Displays the starting position of the field.

Size Displays the size in bytes of the field.

Type Use to specify the field's type, such as CHAR or INT.

Valid Line Commands:

O Use to open this record to its next level.

Valid Option Commands:

R Use to refresh the list. If you have added any records, they are now displayed in the list.

- d. If this is not a default record, you can add or delete fields, or edit existing fields. In that case, IBM File Export opens the Field List (ABX\$ZFDL) panel.

```
ABX$ZFDL V1R1 ----- Field List ----- 2004/11/10 10:47:27
Option ==>                               Scroll ==> PAGE

Repository HLQ
Parent Record                               >
                                           Row 1 of 2

Cmd  Field                               Type

Valid Line Commands : E - Edit D - Delete
Valid Option Commands: S - Add Simple Field G - Add Group Field R - Refresh
```

Figure 43. Field List (ABX\$ZFDL) panel

Panel Fields:

Repository HLQ

Displays the repository's high-level qualifier.

Parent Record

Displays the name of the parent record.

Field Displays the names of the fields that have been defined for this record.

Type Displays whether this is a **SIMPLE** or **GROUP** record.

Valid Line Commands:

- E** Specify to edit an entry in this list.
- D** Specify to delete an entry from the list.

Valid Option Commands:

- S** Specify to add a simple field to the list.
- G** Specify to add a group field to the list.
- R** Use to refresh the list. If you have added any records, they are now displayed in the list.

- e. If you specify **E** to edit an entry on the list, and if this is a SIMPLE field, the Edit Simple Field (ABX\$ZESF) panel opens. On this panel, you can update the entries for any text entry field on the panel. When you have completed your work, press **Enter** to continue.

```
ABX$ZESF V1R1 ----- Edit Simple Field ----- 2004/11/11 10:51:15
```

```
Option ==>
```

```
More: +
```

```
Parent Record
```

```
>
```

```
Field Name
```

```
Enter Field information below:
```

```
Field Position
```

```
Field Length
```

```
Field Type
```

```
Occurs
```

```
Depending On
```

```
Precision
```

```
Scale
```

```
Signed          N (Y/N)
```

```
Scramble        N (Y/N)
```

```
Group Length
```

```
Maximum Length
```

```
Position Algorithm
```

```
Group Length Algorithm
```

```
Enter "/" to select option
```

```
Specify additional Position Algorithm
```

```
Specify additional Group Length Algorithm
```

Figure 44. Edit Simple Field (ABX\$ZESF) panel

Panel Fields:

Parent Record

Displays the name of the parent record to which this field belongs.

Field Name

Displays the name of the field you want to edit.

Field Position

Specify the position of the field in the record. Position can be specified:

- As an integer type constant like 1 or 100.
- Relative to the end of the record, such as 'end - 10', which indicates ten bytes from the end of the record.
- As a hex constant.

Field Length

Displays the length in bytes of the field.

Type Specify the field's type, such as CHAR or INT.

Occurs

Specify an integer that indicates the number of times a variable length repeating group occurs.

Depending On

Specify the object of the Occurs. If the number of times a repeating group can occur in the record depends on this object of the Occurs, that dependency should be specified here. For information about Occurs Depending On, see "Understanding Occurs Depending On" on page 262.

Precision

Specify the precision of numeric data.

Scale Specify the scale of numeric data.

Signed

Specify whether (Y) or not (N) the numeric data is signed.

Scramble

Specify whether (Y) or not (N) the field should be scrambled.

Group Length

Specify the length of the group.

Maximum Length

Specify the maximum length of the field.

Position Algorithm

Because the length of a group item containing variable length data is not a constant, an algorithm is required to express the position of a field, and in some cases, the length of the field or group item. Specify the position algorithm in this field.

Group Length Algorithm

Because the length of a group item containing variable length data is not a constant, an algorithm is required to express the position of a field, and in some cases, the length of the field or group item. Specify the position algorithm in this field.

Enter "/" to select option

To add or delete a position algorithm, group length algorithm, or both, type a slash (/) in the appropriate field or fields. The Algorithm List (ABX\$ZALG) panel opens.

- f. If you specify **E** to edit an entry on the list, and if this is a GROUP field, the Edit Group Field (ABX\$ZEGF) panel opens. On this panel, you can update the entries for any field on the panel.

```

ABX$ZEGF V1R1 ----- Edit Group Field ----- 2004/11/11 10:57:53

Option ==>

More:      +

Parent Record      >

Field Name

Subrecord          >

Enter Field information below:

Field Position

Field Length

Field Type          (REDEFINE/SUBREC)

Occurs

Depending On

Scramble           N (Y/N)

Group Length

Maximum Length

Position Algorithm

Group Length Algorithm

Enter "/" to select option

Specify additional Position Algorithm

Specify additional Group Length Algorithm

Valid Option Commands:  E - Edit Subrecord  O - Open Subrecord

```

Figure 45. Edit Group Field (ABX\$ZEGF) panel

Panel Fields:

Parent Record

Displays the name of the parent record to which this field belongs.

Field Name

Displays the name of the field to which this subfield belongs.

Field Name

Displays the name of the subrecord you want to edit.

Field Position

Specify the position of the field in the record. Position can be specified:

- As an integer type constant like 1 or 100.
- Relative to the end of the record, such as 'end - 10', which indicates ten bytes from the end of the record.
- As a hex constant.

Field Length

Displays the length in bytes of the field.

Field Type

Specify **REDEFINE** or **SUBREC**. **REDEFINE** specifies that you are defining more than one linked record. **SUBREC** specifies that you are defining a single record.

Occurs

Specify an integer that indicates the number of times a variable length repeating group occurs.

Depending On

Specify the object of the Occurs. If the number of times a repeating group can occur in the record depends on this object of the Occurs, that dependency should be specified here. For information about Occurs Depending On, see "Understanding Occurs Depending On" on page 262.

Scramble

Specify whether (Y) or not (N) the field should be scrambled.

Group Length

Specify the length of the group.

Maximum Length

Specify the maximum length of the field.

Position Algorithm

Because the length of a group item containing variable length data is not a constant, an algorithm is required to express the position of a field, and in some cases, the length of the field or group item. Specify the position algorithm in this field.

Group Length Algorithm

Because the length of a group item containing variable length data is not a constant, an algorithm is required to express the position of a field, and in some cases, the length of the field or group item. Specify the position algorithm in this field.

Enter "/" to select option

To add or delete a position algorithm, group length algorithm, or both, type a slash (/) in the appropriate field or fields. The Algorithm List (ABX\$ZALG) panel opens.

Valid Line Commands:

- E** Use to edit the subrecord.
- O** Use to open a list of fields that comprise this subrecord.

From the Edit Group Field panel, you can choose to:

- Edit the Subrecord by typing **E** on the Option line. The Record Layouts Selection panel opens. Choose the subrecord you want to edit by typing an **S** on its Cmd line. This opens the appropriate Edit *xxxxx* Field panel (either *ABX\$ZESF* or *ABX\$ZEGF*), where *xxxxx* is either *SIMPLE* or *GROUP*, where you can perform your edits.

- Open the subrecord by typing **O** on the Option line. The Field List (ABX\$ZFDL) panel opens (see Figure 43 on page 79) so that you can continue your record layout definition.

When you have completed your work, press **Enter** to continue.

- g. If you specify **S** on the Option line to add a SIMPLE field, the Create Simple Field (ABX\$ZESF) panel opens. When you have completed your work, press **Enter** to continue.

```

ABX$ZCSF V1R1 ----- Create Simple Field ----- 2004/11/11 10:51:15

Option ==>

                                     More:   +

Parent Record                       >

Field Name

Enter Field information below:

Field Position

Field Length

Field Type

Occurs

Depending On

Precision

Scale

Signed          N (Y/N)

Scramble        N (Y/N)

Group Length

Maximum Length

Position Algorithm

Group Length Algorithm

Enter "/" to select option

Specify additional Position Algorithm

Specify additional Group Length Algorithm

```

Figure 46. Create Simple Field (ABX\$ZCSF) panel

Panel Fields:

Parent Record

Displays the name of the parent record to which this field belongs.

Field Name

Displays the name of the field you want to edit.

Field Position

Specify the position of the field in the record. Position can be specified:

- As an integer type constant like 1 or 100.
- Relative to the end of the record, such as 'end - 10', which indicates ten bytes from the end of the record.
- As a hex constant.

Field Length

Displays the length in bytes of the field.

Type Specify the field's type, such as CHAR or INT.

Occurs

Specify an integer that indicates the number of times a variable length repeating group occurs.

Depending On

Specify the object of the Occurs. If the number of times a repeating group can occur in the record depends on this object of the Occurs, that dependency should be specified here. For information about Occurs Depending On, see "Understanding Occurs Depending On" on page 262.

Precision

Specify the precision of numeric data.

Scale Specify the scale of numeric data.

Signed

Specify whether (Y) or not (N) the numeric data is signed.

Scramble

Specify whether (Y) or not (N) the field should be scrambled.

Group Length

Specify the length of the group.

Maximum Length

Specify the maximum length of the field.

Position Algorithm

Because the length of a group item containing variable length data is not a constant, an algorithm is required to express the position of a field, and in some cases, the length of the field or group item. Specify the position algorithm in this field.

Group Length Algorithm

Because the length of a group item containing variable length data is not a constant, an algorithm is required to express the position of a field, and in some cases, the length of the field or group item. Specify the position algorithm in this field.

Enter "/" to select option

To add or delete a position algorithm, group length algorithm, or both, type a slash (/) in the appropriate field or fields. The Algorithm List (ABX\$ZALG) panel opens.

- h. If you specify **G** on the Option line to add a GROUP field, the Create Group Field (ABX\$ZEGF) panel opens. When you have completed your work, press **Enter** to continue.

ABX\$ZCGF V1R1 ----- Create Group Field ----- 2004/11/11 10:57:53

Option ==>

More: +

Parent Record >

Field Name >

Subrecord >

Enter Field information below:

Field Position

Field Length

Field Type (REDEFINE/SUBREC)

Occurs

Depending On

Scramble N (Y/N)

Group Length

Maximum Length

Position Algorithm

Group Length Algorithm

Enter "/" to select option

Specify additional Position Algorithm

Specify additional Group Length Algorithm

Figure 47. Create Group Field (ABX\$ZCGF) panel

Panel Fields:

Parent Record

Displays the name of the parent record to which this field belongs.

Field Name

Displays the name of the field to which this subfield belongs.

Field Name

Type the name of the subrecord you want to create.

Field Position

Specify the position of the field in the record. Position can be specified:

- As an integer type constant like 1 or 100.

- Relative to the end of the record, such as 'end - 10', which indicates ten bytes from the end of the record.
- As a hex constant.

Field Length

Displays the length in bytes of the field.

Field Type

Specify **REDEFINE** or **SUBREC**. **REDEFINE** specifies that you are defining more than one linked record. **SUBREC** specifies that you are defining a single record.

Occurs

Specify an integer that indicates the number of times a variable length repeating group occurs.

Depending On

Specify the object of the Occurs. If the number of times a repeating group can occur in the record depends on this object of the Occurs, that dependency should be specified here. For information about Occurs Depending On, see "Understanding Occurs Depending On" on page 262.

Scramble

Specify whether (Y) or not (N) the field should be scrambled.

Group Length

Specify the length of the group.

Maximum Length

Specify the maximum length of the field.

Position Algorithm

Because the length of a group item containing variable length data is not a constant, an algorithm is required to express the position of a field, and in some cases, the length of the field or group item. Specify the position algorithm in this field.

Group Length Algorithm

Because the length of a group item containing variable length data is not a constant, an algorithm is required to express the position of a field, and in some cases, the length of the field or group item. Specify the position algorithm in this field.

Enter "/" to select option

To add or delete a position algorithm, group length algorithm, or both, type a slash (/) in the appropriate field or fields. The Algorithm List (ABX\$ZALG) panel opens.

2. Follow these steps from a Cmd line of the Storage List (ABX\$ZSTL) panel:
 - a. Type an **O** on the Cmd line of a sequential or VSAM storage structure. The Record Layouts (ABX\$ZFRL) panel specific to sequential or VSAM opens.

```

ABX$ZFRL V1R1 ----- Record Layouts ----- 2004/11/10 10:25:14
Option ==>                               Scroll ==> PAGE

Repository
Parent Storage                            >
Parent File                               >
Record Name                               >

                Suppress Default Layouts

                                                Row 1 of 1

Cmd Record Name

Valid Line Commands : U - Unlink M - Manage O - Open
Valid Option Commands: N - New R - Refresh L - Link

```

Figure 48. Record Layouts (ABX\$ZFRL) panel

Panel Fields:

Repository HLQ

Displays the repository's high-level qualifier.

Parent Storage

Displays the name for this storage structure that you chose on the Create Storage panel; for example, ABXSEQ.

Parent File

Displays the name of the file to which this record belongs.

Record Name

Acts as a filter to limit the display at the bottom of the panel. If you enter a value here, you must type R on the Option line to refresh the display.

Suppress Default Layouts

Place a slash on the line preceding this field if you want to filter the display to omit showing defaults. You will need to refresh the display for the choice to take effect.

Record Name

Displays the names of any records that have already been defined.

Valid Line Commands:

- U** Use to unlink (remove) a record from the list.
- M** Use to add selection criteria or define a relationship specification for the selected record.
- O** Use to open the record to lower-level subrecord.

Valid Option Commands:

- N** Use to add a record to the list. You will also need to specify a name for the record in the field provided at the top of the panel.
- R** Use to refresh the list. If you have added any records, they are now displayed in the list.

- L Use to link (add) another record to the list. A list of records available to be linked will be displayed.
- b. Type an **O** on the Cmd line next to a record. If this record is a default record, the fields that comprise it cannot be changed. IBM File Export takes you directly to the Field List panel. The Field List (ABX\$ZFDD) panel opens (see Figure 42 on page 78). If this is not a default record, you can add or delete fields, or edit existing fields. In that case, IBM File Export opens the Field List (ABX\$ZFDL) panel (see Figure 43 on page 79).
- c. Type **U** on the Cmd line next to a record to unlink (remove) that record from the list of records comprising the segment.
- d. Type **M** on the Cmd line next to a record to add or delete selection criteria for that record or to define relationship specifications for the record. The Manage Record Layouts panel opens.

```

ABX$ZMRL V1R1 ----- Manage Record Layouts ----- 2004/12/22 11:54:13
Option ==>

Repository HLQ
Parent Storage >
Parent File >
Current Record >

Data specification
 1 Edit Selection Criteria
 2 Relationship Specification

```

Figure 49. Manage Record Layouts (ABX\$ZMRL) panel

Panel Fields:

Repository HLQ

Displays the repository’s high-level qualifier.

Parent Storage

Displays the name for this storage structure that you chose on the Create Storage panel; for example, ABXVSM1.

Parent File

Displays the name of the parent storage structure to which this record belongs.

Parent File

Displays the name of the parent file to which this record belongs.

Current Record

Displays the name of the current record.

Valid Line Commands:

- 1 Displays a panel on which you can define selection criteria
- 2 Displays a panel on which you can begin to define record relationships.

On this panel, selecting Option 1 takes you to the Selection Criteria (ABX\$ZSCR) panel. Selecting Option 2 takes you to the Manage Record Relations (ABX\$ZREL) panel. To understand the panel flow to define record relationships, see “Managing Record Relationships” on page 92.

- e. When you select Option 1, the Selection Criteria (ABX\$ZSCR) panel opens. On this panel you can add or delete selection criteria for the record. A selection criterion typically is entered as a WHERE clause. Note that, when specifying a value to be matched in the WHERE clause, that value is case-sensitive. If you enter state='Alabama', IBM File Export will not select records where state='alabama'.

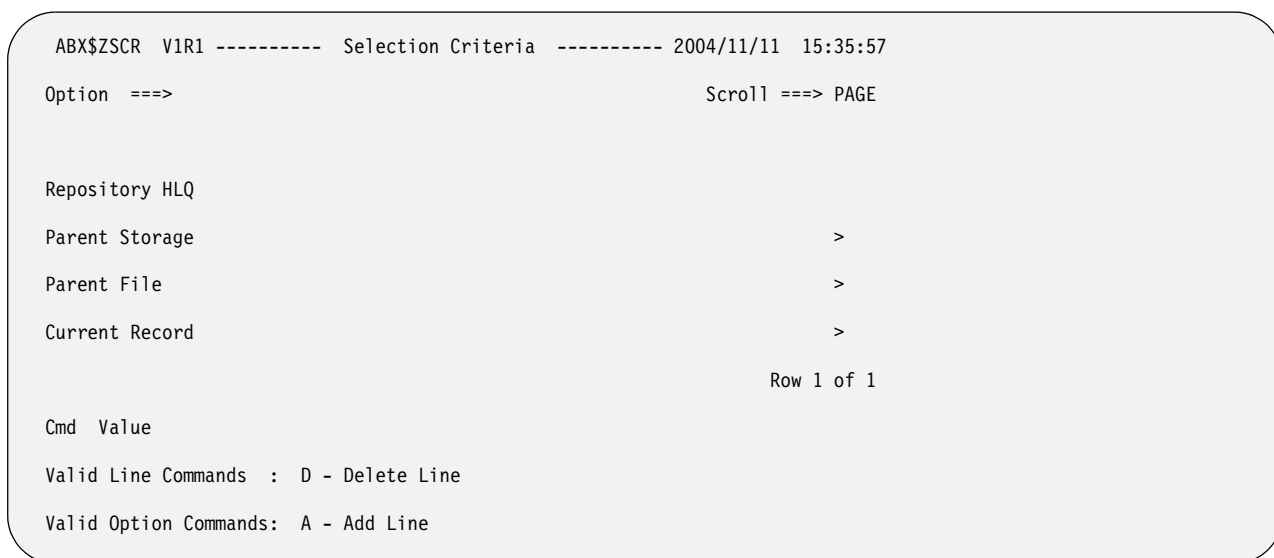


Figure 50. Selection Criteria (ABX\$ZSCR) panel

Panel Fields:

Repository HLQ

Displays the repository’s high-level qualifier.

Parent Storage

Displays the name for this storage structure that you chose on the Create Storage panel; for example, ABXVSM1.

Parent File

Displays the name of the file to which this record belongs.

Record Name

Displays the name of the current record.

Cmd Enter any valid line command in this field.

Value

Type the value to be matched as the selection criteria, typically in the form of a WHERE clause.

Valid Line Commands:

D Deletes the selected value from the list.

Valid Option Commands:

A Adds a blank line on which you can enter a value to be matched.

- f. Type **O** on the Cmd line next to a record to open that record’s list of fields. If this record is a default record, the fields that comprise it cannot be changed. IBM File Export takes you directly to the Field List panel. The Field List (ABX\$ZFDD) panel opens (see Figure 42 on page 78). If this is not a default

record, you can add or delete fields, or edit existing fields. In that case, IBM File Export opens the Field List (ABX\$ZFDL) panel (see Figure 43 on page 79).

- g. Type **N** on the Option line to add a record. The Create Record panel (ABX\$ZREC, see Figure 22 on page 51) opens.
- h. Type **R** on the Option line to refresh the panel. The panel will be updated with any additions or deletions since the last refresh.
- i. Type **L** on the Option line to link additional records to the segment. The Record Layouts Selection (ABX\$ZRDS) panel opens. Choose the record you want to link by typing **S** on the Cmd line next to its name.

```
ABX$ZRDS  V1R1  -----  Record Layouts Selection  -----  2004/11/11  15:43:02
Option  ==>>                               Scroll ==>> PAGE

Repository HLQ
Record Name                               >
                Suppress Default Layouts
                                                Row 1 of 2

Cmd  Record Name
Valid Line Commands  :  S - Select
Valid Option Commands:  R - Refresh
```

Figure 51. Record Layouts Selection (ABX\$ZRDS) panel

Panel Fields:

Repository HLQ

Displays the repository's high-level qualifier.

Record Name

Displays the name of the current record.

Cmd Enter any valid line command in this field.

Suppress Default Layouts

Place a slash on the line preceding this field if you want to filter the display to omit showing defaults. You will need to refresh the display for the choice to take effect.

Record Name

Displays a list of records from which you can choose records to be added to the segment.

Valid Line Commands:

S Selects the record to be added.

Valid Option Commands:

R Refreshes the list to display any added record layouts.

3. When you have completed your work, pressing **PF3** multiple times will take you back through the panels you have used. You may continue to work with other sequential or VSAM storage structures or continue to press **PF3** to exit the repository.

Managing Record Relationships

Selecting Option 2 on the Manage Record Layouts (ABX\$ZMRL) panel permits you to define relationships between records.

1. When you select Option 2, the Record Relations panel opens.

```
ABX$ZREL  VIRI ----- Record Relations ----- 2004/12/22 12:39:16
Option  ===>                               Scroll  ===> PAGE

Repository HLQ
Current Storage                               >
Current Record                               >

Cmd  Relation Name
```

Figure 52. Record Relations (ABX\$ZREL) panel

Panel Fields:

Repository HLQ

Displays the repository's high-level qualifier.

Current Storage

Displays the name for this storage structure that you chose on the Create Storage panel.

Current File

Displays the name of the file to which this record belongs.

Current Record

Displays the name of the current record for which you are specifying relationship information.

Relation Name

Displays a list of existing previously-defined relationships.

Valid Line Commands:

S Selects the record to be added.

Valid Line Commands:

E Specify to edit an existing relationship.

D Specify to delete an existing relationship.

Valid Option Commands:

AP Specify to add a parent record for the current record.

AP Specify to add a child record for the current record.

R Specify to refresh the list of relationships

2. If you choose to edit an existing relationship, add a parent record for the current record, or add a child record for the current record, the Edit Relation (ABX\$ZERL) panel opens.

```

ABX$ZERL V1R1 ----- Edit Relation ----- 2004/12/22 12:49:57
Option ==>

Repository HLQ

Parent File >
Parent Record >
Child File >
Child Record >

Relation Name >

Data specification
 1 Select Record From List
 2 Relationship Specification

```

Figure 53. Edit Relation (ABX\$ZERL) panel

Panel Fields:

Repository HLQ

Displays the repository’s high-level qualifier.

Parent File

Displays the name of the parent file when you have selected it from the list of files.

Parent Record

Displays the name of the parent record you have selected.

Child File

Displays the name of the child file when you have selected it from the list of files.

Child Record

Displays the name of the child record you have selected.

Relation Name

Specify a name to identify the relation you are defining.

Valid Option Commands:

- 1 Specify to select a record from a list of records displayed for you.
 - 2 Specify to define the relationship between the records..
3. Progress through the panels to define or select parent and child records by selecting Option 1 on this panel. Note that you must complete the panels related to Option 1 before you will be permitted to select Option 2. As you move through the selection process, you will be asked to select:
- a storage structure,
 - a file,
 - and a record layout
- for the record (either child or parent) that you are defining.

4. Once you have defined the repository layouts for both the parent and child records, you may select Option 2 from the Edit Relation (ABX\$ZERL) panel.
5. Selecting Option 2 displays the Relation Field List (ABX\$FRE) panel.

```

ABX$ZFRE  V1R1  -----  Relation Field List  -----  2004/12/22  13:09:03
Option  ===>                               Scroll  ===>  PAGE

Parent Record  TESTREQ                       >
Child Record   TESTREQ                       >

Relation Name  CSREL                          >

Cmd  Child Field Name          Parent Field Name
Valid Line Commands :  E - Edit Relation Field  D - Delete Relation Field
Valid Option Command:  A - Add New Relation Field

```

Figure 54. Relation Field List (ABX\$ZFRE) panel

Panel Fields:

Parent Record

Displays the name of the parent record you have selected.

Child Record

Displays the name of the child record you have selected.

Relation Name

Specify a name to identify the relation you are defining. If you are editing an existing relation, its name appears here.

Valid Line Commands:

E Specify to edit an existing relationship field.

D Specify to delete an existing relationship field.

Valid Option Command:

A Specify to add a new relationship field.

This panel displays a list of previously-defined parent-child relationships. On this panel, you can choose to edit an existing parent-child relationship, delete an existing relationship, or create a new relationship. If you choose to edit an existing relationship or create a new relationship, the Edit Relation Fields (ABX\$ZREE) panel opens to permit you to do so.

6. When you have completed your relationship definition activities, you can press PF3 multiple times to exit the repository.

Specifying DB2 table and RI definitions

Follow these steps to specify DB2 table and RI definitions:

1. Type an **I** on the Option line of a Table List panel. (For more information about the Table List panel, see page 46).
2. The Import DB2 Definitions (ABX\$DBRI) panel opens. This panel permits you to import from the DB2 catalog into the repository those DB2 table definitions and RI definitions you will identify. You may specify or provide a pattern for the table creator and table name. You may optionally specify the number of RI "levels" to be searched.

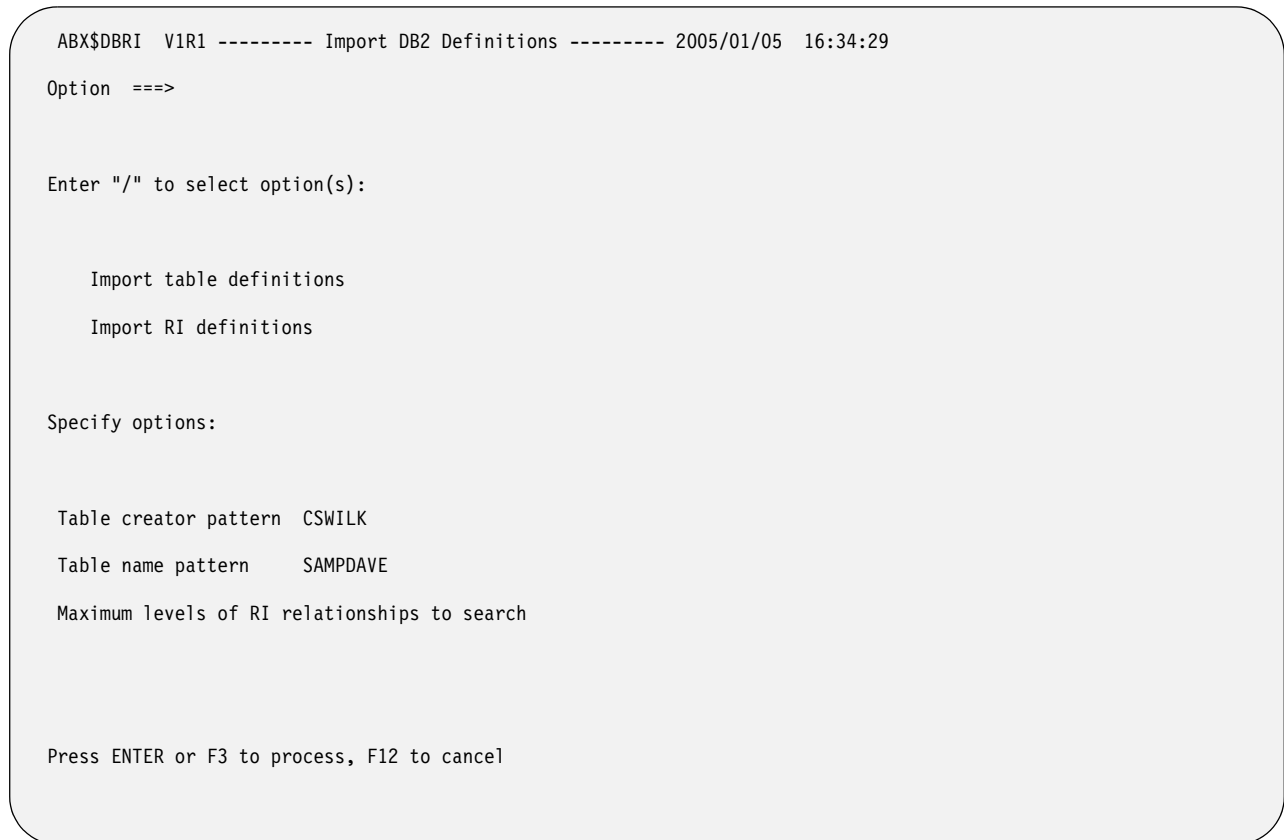


Figure 55. Import DB2 Definitions (ABX\$DBRI) panel

Panel Fields:

Import table definitions

Placing a slash on this line imports your DDL definitions from the DB2 catalog for the table creator and table name (or pattern) you specify.

Import RI definitions

Placing a slash on this line imports the syntax that defines the parent/child relationships for the table creator and table name (or pattern) you specify.

Table creator pattern

Specify a value to identify the table creator that is to be used to retrieve the selected definitions. The value you enter in this field will be used as a pattern and may contain a wild card. Note that if you use a wild card, all tables satisfying the pattern will be retrieved, regardless of their relationship to each other. If you want a specific table that is related to another, to ensure that it is retrieved, specify its table owner here.

Table name pattern

Specify a value to identify the table name that is to be used to retrieve the selected definitions. The value you enter in this field will be used as a pattern and may contain a wild card. Note that if you use a wild card, all

tables satisfying the pattern will be retrieved, regardless of their relationship to each other. If you want a specific table that is related to another, to ensure that it is retrieved, specify its fully qualified name here.

Maximum levels of RI relationships to search

Specify the number of "levels" of RI relationships to be imported into the repository. For example, if A, the parent, is related to B, the child, to find both A and B, you would need to import two levels. Note that B could also have a relationship to another child, C, which is on the same level as B and thus, specifying 2 in this field would import A, B, and C. The value in this field is essentially the "degree of separation" from the topmost parent record to the lowest-level related child record or records.

3. Once IBM File Export has queried the catalog to find all tables matching the wildcard pattern you have specified, the Table List (ABX\$ZTLS) panel is redisplayed with the related tables. You can now open the record layouts for the panels that have been displayed and continue to work in the repository as before. For information about working with record layouts, see "Working with record layouts" on page 49.

Information about the use of the functions that IBM File Export provides in support of DB2 relational integrity (RI) is provided in greater detail in the *IBM File Export for z/OS: User's Guide* in the chapter entitled "Working with the IBM File Export Repository."

Using the parser

You may use either COBOL or PL/I to parse the contents of your repository. The parser translates record layouts from the source input and creates them in the repository. You will first need to configure your parser by following these steps:

1. From the Record Layouts (ABX\$ZRDL) panel, type **P** on the Option line to invoke the Parser Configuration panel.

```
ABX$ZPRS V1R1 --- Parser configuration --- 2004/11/09 14:39:22
```

```
Option ==>
```

```
Enter top-level record name:
```

```
Enter data set and member name for source module:
```

```
Data set name
```

```
Member name          (Blank for member list)
```

```
Language selection
```

```
/ PL/I
```

```
  Cobol
```

```
Compiler options:
```

```
Additional options
```

```
1 INCLUDEs
```

Figure 56. Parser Configuration (ABX\$ZPRS) panel

Panel Fields:

Enter top level record name

Type the name that will be assigned to the object that is being parsed.

Data Set Name

Type the name of the data set that contains the source module that is being parsed. You may browse for the data set name by typing a high level qualifier and an asterisk.

Member Name

Type the name of the member that contains the source module that is being parsed. To browse for the name, leave the field blank to see a member list.

Language Selection

Choose PL/I or COBOL by placing a slash on the appropriate line.

Compiler Options

Type any compiler options on the line provided.

Valid Option Commands:

- 1 Type 1 on the Option line to include additional partitioned data sets or copy books.
2. Complete the fields on this panel. If you want to include additional data sets or copybooks, type 1 on the Option line to display panel ABX\$DSSL. Use this panel to add or delete objects from the INCLUDE list.

When you have completed your work on the Parser Configuration panel, press **Enter**. The COBOL or PL/I source that was specified is displayed in a temporary data set.

You can edit the source in that data set without affecting the base source input. When you press **PF3** from this temporary data set, IBM File Export runs the parser and the record layout that was specified in the top-level record name field on the Parser Configuration panel is translated from the source input and created in the repository.

Part 5. Working with IBM File Export

Chapter 6. Getting Started with the IBM File

Export ISPF Interface	101
Exporting data using IBM File Export for z/OS.	101
What sources and targets can I use?.....	101
Using the Session Management (ABX\$SMNG) panel	102
Specifying selection criteria	103

Chapter 7. Exporting DB2 Data

Chapter 7. Exporting DB2 Data	105
Summary	105
Specify DB2 product options	105
Select a DB2 session	106
Specify the subsystems and LOADLIBs for your source and target	107
Define Relationships	109
Specify and manipulate query data	113
Edit source tables	115
Edit source columns	116
Edit selection criteria	118
Modify target columns	119
Edit query relationships	122
Edit Query Criteria	123
Specify target table information	124
Specify intermediate data set DD statements ...	126
Generate JCL	128

Chapter 8. Exporting IMS data

Chapter 8. Exporting IMS data	133
Select the IMS source and target types	133
Summary	134
The IMS Menu	135
Specify IMS session settings	136

Specify source database	140
If your source processing option is IMS or DLI	141
If your source processing option is DEDB.....	142
If your source processing option is DBB or BMP	143
Specify target database	144
If your target processing option is IMS or DLI..	144
If your target processing option is DEDB, DBB, or BMP.....	146
If your target processing option is FEUNLOAD	147
Define variables and literals	148
Define relationships	150
Define segment specification and manipulation	154
Create DD statements	166
Generate JCL	170

Chapter 9. Exporting Sequential Data or VSAM

Chapter 9. Exporting Sequential Data or VSAM Data	173
Summary	173
The Sequential and VSAM Menu	174
Specify session settings	175
Specify source and target data sets	178
Define variables and literals	181
Define relationships	183
Define field specification and manipulation ...	187
Define type and select statement specification and manipulation	191
Generate JCL	198

This part contains information on using IBM File Export for z/OS.

Chapter 6. Getting Started with the IBM File Export ISPF Interface

IBM File Export provides a TSO/ISPF interface as well as a batch JCL interface. Use the TSO/ISPF panels for "fill-in-the-blanks" identification of sources and targets, and the automatic generation of IBM File Export processing instructions. This chapter provides information about initiating a IBM File Export session and using the ISPF interface to create, view, and work with a session.

Exporting data using IBM File Export for z/OS

To export data using IBM File Export for z/OS, follow this progression:

1. Define one or more sources where the data that you want to export is located. You can define a primary source, a secondary source and relationships between the sources.
2. Optionally, specify any data transformations that are to be applied to your source data as it is exported.
3. Define one or more targets where the data that you export will be located.
4. Generate the data.
5. Load the data at the target location.

Begin your session with the IBM File Export Session Management panel.

What sources and targets can I use?

The table that follows identifies the combinations of sources and targets that are valid when using IBM File Export.

Table 1. Valid Sources and Targets

Valid Sources	Valid Targets								
	DB2	SEQ.	VSAM	IMS	DEDB	DLI	DBB	BMP	FEUnload
DB2	X								
Sequential		X	X						
VSAM		X	X						
IMS				X	X	X	X	X	
DEDB				X	X	X	X	X	
DLI				X	X	X			X
DBB				X	X		X		X
BMP				X	X			X	X
FEUnload									

Notes:

1. You can specify multiple sources as input only when using FEUNLOAD as output.
2. FEUNLOAD is never supported as a source input.

Using the Session Management (ABX\$SMNG) panel

From the Session Management (ABX\$SMNG) panel, you can choose the type of session you want to create, open an existing session, or view a list of existing sessions from which you can select the session to work with. The Session Management panel is discussed in detail in Chapter 6, “Getting Started with the IBM File Export ISPF Interface,” on page 101.

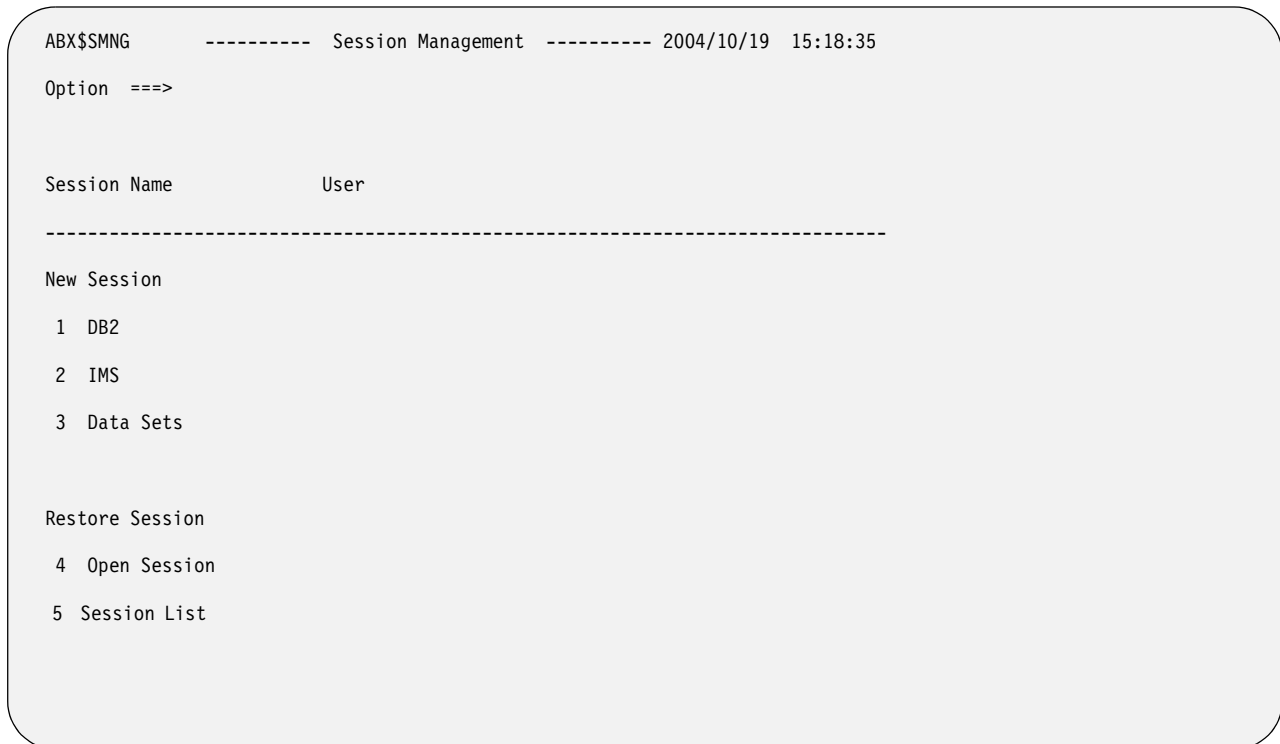


Figure 57. IBM File Export Session Management (ABX\$SMNG) panel

Panel fields:

- 1 Specifying **1** starts a new DB2 session. You must also specify a session name in the **Session Name** field. Panel ABX\$DB2W opens.
- 2 Specifying **2** starts a new IMS session. You must also specify a session name in the **Session Name** field. Panel ABX\$IMST opens.
- 3 Specifying **3** starts a new sequential or VSAM session. You must also specify a session name in the **Session Name** field. Panel ABX\$SEQW opens.
- 4 Specifying **4** starts an existing session whose name you have specified in the **Session Name** field.
- 5 Specifying **5** displays list of existing sessions. If you have included anything in the **Session Name** field, it is ignored.

Specifying selection criteria

Note that, when specifying selection criteria, you will want to ensure that you exactly enter the criterion to be matched. The matching process is case sensitive. That is, if you specify `state='Alabama'`, IBM File Export will not select records where `state='alabama'`.

Chapter 7. Exporting DB2 Data

This chapter explains how to use IBM File Export for z/OS to export DB2 data. The panels that are used in this process and their associated options, fields, and line commands are also explained. Note that the panels illustrated in this book are representative of the panels in the product and may differ slightly from the product panels.

Summary

Specify DB2 product options

“Specify DB2 product options”

Select a DB2 session

“Select a DB2 session” on page 106

Define the source and target DB2 subsystems, including the DB2 LOADLIBs and plan names for the user interface and batch utility

“Specify the subsystems and LOADLIBs for your source and target” on page 107

Specify external relationships

“Define Relationships” on page 109

Define queries and specify manipulation of query data

“Specify and manipulate query data” on page 113

Generate JCL

“Generate JCL” on page 128

Specify DB2 product options

If you will use IBM File Export with DB2 databases, follow these steps to specify DB2 options for your product.

1. To specify DB2 options, type **D** on the command line of the Settings (ABX\$GOPT) panel and press **Enter**. The DB2 Product Settings (ABX\$DOPT) panel opens.

```

ABX$DOPT      ----- DB2 Product Settings ----- 2004/06/18 09:18:12
Option ==>
Sort Job Statement Information:
==> //JOB CARD JOB MYNAME,CLASS=A,NOTIFY=&SYSUID
==> /*
==> /*
==> /*

DB2 Load Job Statement Information:
==> //JOB CARD JOB MYNAME,CLASS=A,NOTIFY=&SYSUID
==> /*
==> /*
==> /*

```

Figure 58. DB2 Options (ABX\$DOPT) panel

Panel fields:

Sort Jobcard

Type the job card information for the job that will be used for the MVS SORT facility that is used by DB2. You should include a REGION= parameter on this JOB statement.

DB2 Load Jobcard

Type the job card information for the job that will be used for the DB2 LOAD Utility. You should include a REGION= parameter on this JOB statement.

2. Set the defaults for using IBM File Export for z/OS with DB2 databases by filling in the fields on the DB2 Product Settings (ABX\$DOPT) panel. Pressing **PF3** returns you to the Settings (ABX\$GOPT) panel. Pressing **PF3** again takes you to the Main Menu.

Select a DB2 session

From the IBM File Export Session Management panel, ABX\$SMNG (see page 102), select option **1, DB2**, to provide specifications for a DB2 database. You must specify a session name for the session you are creating. The DB2 Menu, ABX\$DB2W, opens.

Note: If you want to work with an existing session, specifying **4** on the Option line of the IBM File Export Session Management (ABX\$SMNG) panel together with a session name in the Session Name field opens the specified session. Specifying **5** on the Option line displays a list of previously-defined sessions from which you can choose the one to work with.


```
ABX$DB2W      ----- DB2 Menu ----- 2004/08/02 15:52:08
Option ==>>
Session Name:
Database specification
  1 Source and Target
Data specification
  2 Relationship Specification
  3 Query Specification and Manipulation
JCL specification
  4 Generate JCL
```

Figure 59. DB2 Menu (ABX\$DB2W) panel

Panel fields:

Session Name

Specifies the name of your current session.

Valid Option Commands:

- 1 Select this option to identify the source and target DB2 subsystems you will use. When you select this option, panel ABX\$DBST opens.
- 2 Select this option to create external relationships to be used during this session. When you select this option, the ABX\$RELS panel opens.
- 3 Select this option to define query information and how queries are constructed and manipulated. When you select this option, panel ABX\$DBQL opens.
- 4 Select this option to display panel ABX\$DBJC on which you specify the information needed to generate the JCL to be used to generate the target output.

Specify the subsystems and LOADLIBs for your source and target

Use the DB2 Source and Target panel (ABX\$DBST) to specify the DB2 subsystem and LOADLIB where the source resides and where you want the target to reside.

- 1. On the DB2 Menu panel, type 1 on the command line and press enter. The DB2 Source and Target (ABX\$DBST) panel opens.

```

ABX$DBST      ----- DB2 Source and Target ----- 2004/08/02 16:02:33
Option ==>>

Session Name:

Source:

  DB2 Subsystem

  DB2 Loadlib

  Plan Name For FE UI

  Plan Name For FE Batch Utility
  Plan Name for Load Executor

Target:

  DB2 Subsystem 2

  DB2 Loadlib

```

Figure 60. DB2 Source and Target (ABX\$DBST) panel

Panel fields:

Session Name

Specifies the name of your current session.

DB2 Subsystem (source)

Type the DB2 subsystem name of the system where the source DB2 database is located.

DB2 Loadlib (source)

Type the name of the source DB2 load library.

Plan Name For FE UI

Type the plan name to be used for the IBM File Export for z/OS user interface.

Plan Name For FE Batch Utility

Type the plan name to be used for the IBM File Export for z/OS batch utility.

Plan Name For Load Executor

Type the plan name for the Load Executor in this field.

DB2 system (target)

Type the DB2 sub-system name of the system where the target DB2 database is located.

DB2 Loadlib (target)

Type the name of the target DB2 load library.

2. On the Source and Target panel (ABX\$DBST), specify the DB2 subsystem and LOADLIB where the source data resides and the subsystem and LOADLIB where you want the target output to reside. You can create a target that is a DB2 table or a flat file. If you want to create a target that is a DB2 table, type the DB2 subsystem and load libraries where the target table or tables that you want to create will reside in the appropriate fields.
3. Type the plan names for the user interface, batch utility, and LOAD Executor in the fields provided.
4. When you have completed your entries, press **PF3** to return to the DB2 Menu.

Define Relationships

Follow these steps to provide information to define external relationships. Before doing so, you may want to review the information about application-defined relationships in “Defining relationships in IBM File Export” on page 5.

1. On the DB2 Menu panel, select option 2, **Relationship Specification**. Panel ABX\$RELS opens.

```
ABX$RELS      ----- Relationship Specification ----- 2004/06/20 14:10:00
Option  ==>                               Scroll ==> PAGE

Session Name

Row 1 of 1

Cmd  Label      Type  Dataset

Valid Line Commands:  E - Edit Entry  D - Delete Entry
Valid Option Commands: AF - Add Filter  AC - Add Content
```

Figure 61. Relationship Specification (ABX\$RELS) panel

Panel fields:

Session Name

Lists the name of the IBM File Export for z/OS session under which you are working.

Panel columns:

Cmd Type any line command available on this panel in this column next to the desired entry.

Label

Lists the name you have assigned to this filter or content specification.

Type Lists an **F** if this is a filter or **C** if this is a content specification.

Data Set

Names the data set with which filter or content relationship data has been associated.

Valid Line Commands:

E Use to edit an entry in this list. When you type the **E** line command in the **Cmd** field, the ABX\$RELF (if this line is for a filter) or ABX\$RELC (if this line is for a content specification) panel opens.

D Use to delete an entry from this list.

Valid Option Commands:

AF Use to add a filter to the current data set. When you type the **AF** option, the ABX\$RELF panel opens.

AC Use to add a content specification to the current data set. When you type the **AC** option, the ABX\$RELC panel opens.

2. If you entered **AF** to add a filter, the Filter Specification panel (ABX\$RELF) opens. Use this panel to add a filter. The name of the source data set that contains the related data that you want to add to your target is displayed in the Source Name field if you have selected that data set from the External Data Relationships list.

Default values appear in the rest of the fields. You can change the value in any field by overtyping it.

Note: External data is that data contained in a data set from which source data will be selected to be exported to the target, as, for example, a VSAM file you want to export into your target table, or a sequential file from which you want to pull fields to be added to your target object.

```
ABX$RELF      ----- Filter Specification ----- 2004/06/20 14:39:58

Source Name
DSN
Position      0
Length        0
Data Type     HEX

Press ENTER or F3 to process, F12 to cancel
```

Figure 62. Filter Specification (ABX\$RELF) panel

Panel fields:

Source Name

Type the name you are assigning to this filter.

DSN Type the name of the data set that is the source for this filter. If you specify a partial qualifier together with an asterisk, panel ABX\$DSL5 opens. On this panel, you can select the data set you want to use.

Position

Type the position of the source data in this field. Position can be specified:

- As a numeric integer type constant like 1 or 100
- Relative to the end of the record like 'end - 10' (ten bytes from the end if the record).
- Using a hex constant (x'22' or h'22') which will always be interpreted as an offset (never as a 'position').

Length

Type the length of the source data in this field. Length is a numeric literal. It is the length of the bytes required in memory for the source data.

Data Type

Type the data type of the source data set in this field. (For a description of these data types, see Table 3 on page 257.) You can specify any of the following types:

- INT
- CHAR
- FLOAT
- HEX
- SHORT

- DOUBLE
- PACKED

Pressing **PF3** returns you to the Relationship Specification panel, ABX\$RELS).

3. If you entered **AC** to add a content specification, the Relation Content panel (ABX\$RELC) opens.

```

ABX$RELC ----- Relation Content ----- 2004/06/20 14:49:23

Content Name
DSN

Key For Content
  Position    0
  Length      0
  Data Type   HEX

Data For Content
  Position    0
  Length      0
  Data Type   HEX

Press ENTER or F3 to process, F12 to cancel
  
```

Figure 63. Relation Content (ABX\$RELC) panel

Panel fields:

Content Name

Type the name you are assigning to this content specification.

DSN Type the name of the data set that is the source for this content specification. If you put an asterisk in this field, panel ABX\$DSLS opens. On this panel, you can select the data set you want to use.

Position

Type the position of the source data in this field. Position can be specified:

- As a numeric integer type constant like 1 or 100
- Relative to the end of the record like 'end - 10' (ten bytes from the end if the record).
- Using a hex constant (x'22' or h'22') which will always be interpreted as an offset (never as a 'position').

Length

Type the length of the content key or content data (as appropriate) in this field. Length is a numeric literal and is the length of the bytes required in memory for the source data.

Data Type

Type the data type of the source data in this field. (For a description of these data types, see Table 3 on page 257.) You can specify any of the following types:

- INT
- CHAR
- FLOAT
- HEX
- SHORT
- DOUBLE
- PACKED

Note: The position, length, and data type fields are repeated first for the key and second for the content specification.

Use this panel to specify information about the content data set. The DSN field is filled in for you if you have selected the data set from the External Data Relationships list. Default values appear in the rest of the fields. You can change the value in any field by overtyping it.

To add content, you select data from a data set and match that data to the data from a specific column in the source in order to populate your target. Use the key to identify the key column. Preferably, that column is unique, as, for example, Account Number.

Use the Data to identify actual data that is to be selected from the source to be written to the target.

To summarize, use a content specification to include data from an external source. Use the Key to specify what data IBM File Export is to compare to the source file, and the Data to specify what data IBM File Export is to include in the target output when the key is found.

For example, assume that your source file contains addresses using two-character State abbreviations. But you would rather have addresses with the State name spelled out in the target data. To accomplish this, you would define a Content Relationship in File Export to use an external file that contains the abbreviated State name in columns 1 and 2, and the spelled out State name in columns 3 through 28. In this example, the Key Column in the external file would be the abbreviated State name (since the same data exists in the source file) in columns 1 and 2, and the Data Column that you want to include in the target data would be the data contained in columns 3 through 28. So that when File Export finds 'TX' in the source file as the abbreviated State name, for example, the Key from the external file is matched, and File Export uses the spelled-out State name from the external file to populate the target data file.

Note: External data is that data contained in a data set from which source data will be selected to be exported to the target, as, for example, a VSAM file you want to export into your target table, or a sequential file from which you want to pull fields to be added to your target object.

Pressing **PF3** returns you to the Relationship Specification panel, (ABX\$RELS).

4. Entering **E** on the Cmd line next to a filter or content specification, the ABX\$RELF (if this line is for a filter) or ABX\$RELC (if this line is for a content specification) panel opens. When you have completed your work on this panel, pressing **PF3** returns you to the Relationship Specification panel (ABX\$RELS).
5. Pressing **PF3** again returns you to the DB2 Menu (ABX\$DB2W).

Specify and manipulate query data

1. On the DB2 Menu panel, select Option 3 and press **Enter**. The Queries (ABX\$DBQL) panel opens.

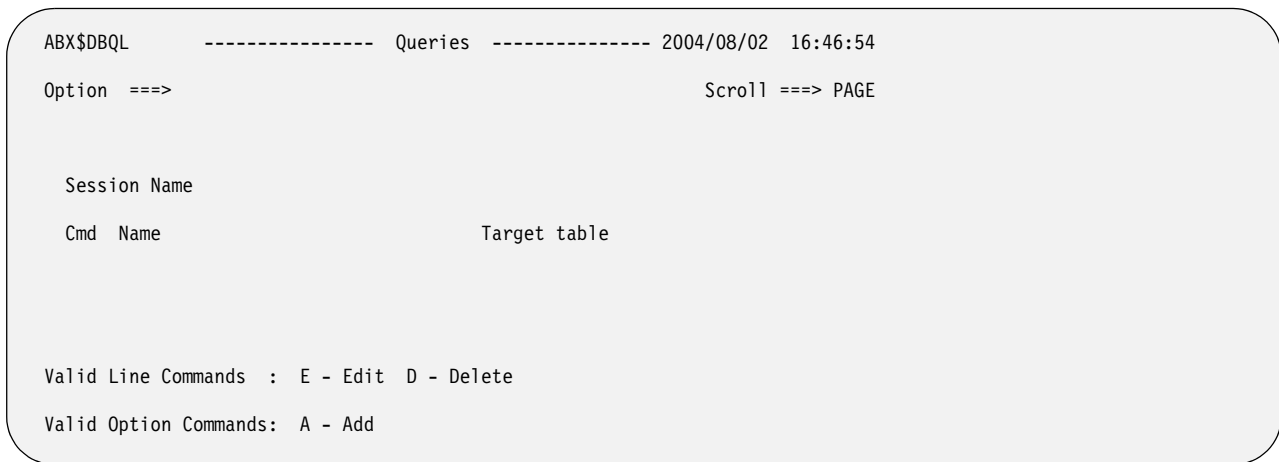


Figure 64. Queries (ABX\$DBQL) panel

Panel Fields:

Session Name

Specifies the name of your current session.

Cmd Type a valid line command in this column next to the desired entry.

Name

This field displays the name under which the query is stored.

Target Table

This field displays the name of the table against which the named query is to be run.

Valid Line Commands:

E Type **E** in the Cmd field to edit the query you have selected.

D Type **D** in the Cmd field to delete the query you have selected from the list of queries.

Valid Option Commands:

A Type **A** on the Option line to define queries to be added to this list.

2. The Queries panel displays a list of previously-defined queries. To edit a query on the list, type **E** on the Cmd line preceding its name. To add a query to the list, type **A** on the Option line. In either case, panel ABX\$DBQE, Edit Query, opens.

ABX\$DBQE V1R1 ----- Edit Query ----- 2005/01/06 14:23:36

Option ==>

Session Name

Query Name

Source SELECT statement

- 1 Tables
- 2 Columns
- 3 Selection criteria

Additional processing

- 4 Column modifications
- 5 Query relationship
- 6 External relationships

Target

- 7 Table
- 8 Intermediate data set

Figure 65. Edit Query (ABX\$DBQE) panel

Panel Fields:

Session Name

Specifies the name of your current session.

Query Name

Specifies the name under which the query is stored.

Valid Option Commands:

- 1 Type 1 on the Option line to specify information about the source table for the data.
- 2 Type 2 on the Option line to specify information about the query itself and the columns from which it selects data.
- 3 Type 3 on the Option line to specify selection criteria for the data that will be selected.
- 4 Type 4 on the Option line to specify how data will be modified as it is written to the target table.

- 5 Type 5 on the Option line to provide information about the parent-child relationship for a query.
 - 6 Type 6 on the Option line to provide one or more WHEN column IN clauses to be used to filter the intermediate result set (the data extracted from DB2).
 - 7 Type 7 on the Option line to specify information about the target table and the data written to it.
 - 8 Type 8 on the Option line to specify the DD parameters that define the intermediate data set.
3. On panel ABX\$DBQE, you will specify information about:
 - The tables, columns, and selection criteria that comprise the SELECT statement that selects data from the source.
 - Additional processing that will occur against the intermediate result set.
 - The target table that will be created and the DD statements that define the intermediate data set.
 4. When you have completed your work on the Edit Query panel, press **PF3** to return to the Queries panel.

Edit source tables

1. If you specified 1 on the Option line of the Edit Query panel, IBM File Export displays the Source Tables panel. The Session Name and Query Name are provided for you on that panel. Use this panel to provide information about the table or tables that will be used to provide source input.

```

ABX$DBTL      ----- Source Tables ----- 2004/08/09  11:37:02
Option  ==>>                               Scroll ==>> PAGE

Session Name

Query Name

Cmd  Owner      Name

Valid Line Commands :  E - Edit  D - Delete
Valid Option Commands:  A - Add
  
```

Figure 66. Source Tables (ABX\$DBTL) panel

Panel Fields:

Session Name

Specifies the name of your current session.

Query Name

Specifies the name of the query for which you are defining the source tables.

Cmd Type any valid line command in this field.

Owner

Displays the name of the owner of the source table.

Name

Displays the name of the source table.

Valid Line Commands:

E Type **E** in the Cmd field to edit the table name or table owner for the source table you have selected.

D Type **D** in the Cmd field to delete the source table you have selected.

Valid Option Commands:

A Type **A** on the Option line to define source tables to be added to the list.

2. To delete a table from the list, type **D** in the Cmd field for that table.
3. To add a table, type **A** on the Option line. To edit a table name or owner name, type **E** in the Cmd field. In either case, a pop-up panel opens requesting the new or revised table name and/or table owner.
4. When you have completed work on this panel, press **PF3** to return to the Edit Query panel.

Edit source columns

1. If you specified **2** on the Option line of the Edit Query panel, IBM File Export displays the Query Columns (ABX\$DBCL) panel. The Session Name and Query Name are provided for you on that panel. Use this panel to provide information about the source input table columns accessed by this query.

```

ABX$DBCL      ----- Query Columns ----- 2004/08/09 15:15:00
Option  ==>>                               Scroll ==>> PAGE

Session Name

Query Name

Row 1 of 1

Cmd Table           Column           Type

Valid Line Commands : D - Delete
Valid Option Commands: A - Add Column L - Add Columns from list
  
```

Figure 67. Query Columns (ABX\$DBCL) panel

Panel Fields:**Session Name**

Specifies the name of your current session.

Query Name

Specifies the name of the query for which you are defining the columns from the source table.

Cmd Type any valid line command in this field.

belonging to the source table. Type **S** in the Cmd field of the table/column combination you want to add to your query.

```

ABX$DBCS      ----- Columns Selection ----- 2004/08/09 15:36:58
Option  ==>                               Scroll ==> PAGE
  Session Name
  Query Name

Cmd  Table                Column          Type          Length

Valid Line Commands : S - Select
  
```

Figure 69. Columns Selection (ABX\$DBCS) panel

Panel Fields:

Session Name

Specifies the name of your current session.

Query Name

Specifies the name of the query for which you are defining the columns from the source table.

Cmd Type any valid line command in this field.

Table Displays the name of the source table.

Column

Displays the name of the table column.

Type Displays the type of the table column (see Table 3 on page 257) which can be:

- INT
- CHAR
- FLOAT
- HEX
- SHORT
- DOUBLE
- PACKED

Length

Displays the length of the column.

Valid Line Commands:

S Type **S** in the Cmd field to select a column from the list to be added to the Query Columns panel.

5. Press **Enter** to return to the Query Columns panel. Press **PF3** on that panel to return to the Edit Query panel.

Edit selection criteria

1. If you specified **3** on the Option line of the Edit Query panel, IBM File Export displays the Selection Criteria panel. The Session Name is provided for you on

that panel. Use this panel to provide information about the selection criteria used to select source input.

2. Panel ABX\$CLST opens.

```
ABX$CLST      ----- Selection Criteria ----- 2004/06/20 13:49:03
Option  ==>                               Scroll ==> PAGE
      Session Name:
                                     Row 1 of 2
Cmd  Value

Valid Line Commands:  D - Delete Line
Valid Option Commands: A - Add Line
```

Figure 70. Selection Criteria (ABX\$CLST) Panel

Panel fields:

Session Name

Displays the name of the session.

Cmd Type any line command available on this panel in this column next to the desired entry.

Value

Type the value to be assigned as a selection criterion, for example, LITERAL= 'X'0100'. Note that selection criteria are case-sensitive. If you specify state='Alabama', IBM File Export will not select records where state='alabama'.

Valid Line Commands:

D Type **D** to delete an entry from the list.

Valid Option Commands:

A Type **A** to add a line on which you can specify additional selection criteria.

3. Add selection criteria to or delete selection criteria from the displayed list.

Modify target columns

1. If you specified 4 on the Option line of the Edit Query panel, IBM File Export displays the Target Columns (ABX\$DBTC) panel. The Session Name and Query Name are provided for you on that panel. Use this panel to provide information about the target output you will create.

```

ABX$DBTC      ----- Target Columns ----- 2004/08/11 13:11:37
Option ==>                               Scroll ==> PAGE

  Session Name

  Query Name

                                                    Row 1 of 1

Cmd Column          Target Column Name Scr  Omit  Ext src  New
  PHONE                                N    N    N      N

Valid Line Commands : E - Edit D - Delete
Valid Option Commands: A - Add

```

Figure 71. Target Columns (ABX\$DBTC) panel

Panel Fields:

Session Name

Displays the name of your current session.

Query Name

Displays the name of the query for which you are defining the target output.

Cmd Type any valid line command in this field.

Column

Displays the name of a source column.

Target Column Name

Type the column name under which the source column data will be stored in the output.

Scramble

Type **Y** to scramble the contents of this column on output. Type **N** to output the contents of the column without scrambling. **N** is the default. When a column is scrambled, the contents will retain the same format as in the source, but the contents will be randomized. This is useful when the source contains sensitive data. The algorithm used for scrambling produces reproducible output. This means that the output of a scrambled value will be the same each time that value is scrambled.

Omit

Type **Y** to omit this column from the target output. Type **N** to include this column in the output. **N** is the default.

Ext src

Displays **Y** to indicate that this column will contain data from an external source (that is, will have an external relationship), or **N** to indicate that the column will not have an external relationship.

New

Displays **Y** to indicate that this column is a new column for the target output, or **N** to indicate that this is column is not new to the target output. **N** is the default.

Valid Line Commands:

E Type **E** in the Cmd field to edit a specific target column.

D Type **D** in the Cmd field to delete a target column.

Valid Option Commands:

- A** Type **A** on the Option line to add a target column.
- 2. If you enter **E** on a Cmd line or **A** on the Option line, the Edit Target Column panel opens.

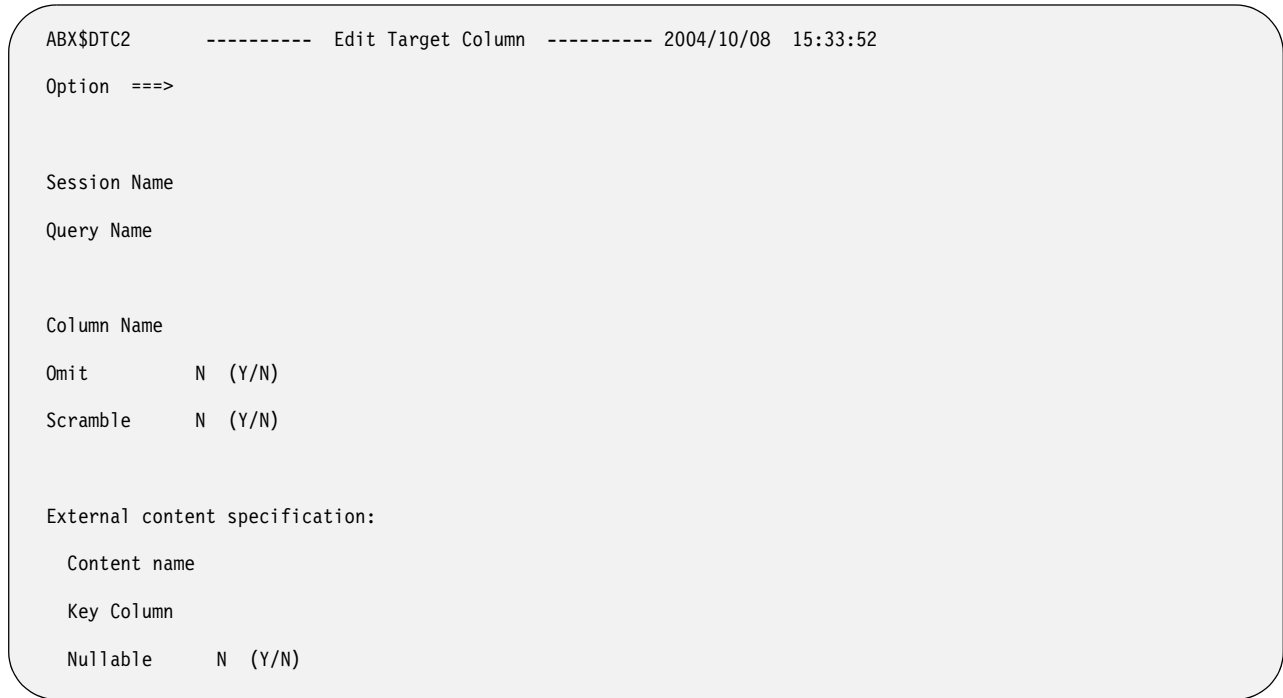


Figure 72. Edit Target Columns (ABX\$DTC1 or ABX\$DTC2) panel

Panel Fields:

Session Name

Displays the name of your current session.

Query Name

Displays the name of the query for which you are defining the target output.

Column Name

Displays the name of a source column.

Target Column Name

Displays the column name under which the source column data will be stored in the output. This field only appears on this panel if you have specified a value for Target Column Name on the Target Columns (ABX\$DBTC) panel.

Omit

Type **Y** to omit this column from the output. Type **N** to include this column in the output. **N** is the default.

Scramble

Type **Y** to scramble the contents of this column on output. Type **N** to output the contents of the column without scrambling. **N** is the default. When a column is scrambled, the contents will retain the same format as in the source, but the contents will be randomized. This is useful when the source contains sensitive data. The algorithm used for scrambling produces reproducible output. This means that the output of a scrambled value will be the same each time that value is scrambled.

Content Name

Type the column name under which this content will be stored in this field.

Key Column

Type the name of the column, taken from the results set, that will be used as a key field.

Nullable

Type **Y** to indicate that this is a nullable column (that is, it can contain null values). Type **N** to indicate that this column is not nullable. **N** is the default.

3. Press **PF3** when you have completed work on this panel to return to the Table Columns panel. Press **PF3** again to return to the Edit Query menu.

Edit query relationships

1. If you specified **5** on the Option line of the Edit Query panel, IBM File Export displays the Edit Relation (ABX\$RRED) panel. This panel permits you to create or edit a relationship that will exist between the statement named in the Statement Name field and the statement you will name in the Parent Statement Name field. Enter the value for the task you want to perform, for example, **1** to select a parent from list, in the Option field. Statement Name represents a table that is an intermediate output that results from the execution of a query.
2. Complete the fields on this panel.

```

ABX$RRED V1R1 ----- Edit Relation ----- 2005/01/06 14:01:54

Option ==>

Session Name

Record Type Name

Statement Name

Parent Statement Name

Data specification

  1 Select Parent From List
  2 Relationship Specification

```

Figure 73. Edit Relation (ABX\$RRED) panel

Panel fields:**Session Name**

Displays the name of the session.

Record Type Name

Displays the name of the record type, such as Debit or Credit.

Statement Name

An identifier of up to 32 characters in length for the parent statement.

Parent Statement Name

Specify an identifier of up to 32 characters in length for the parent statement.

Valid Option Commands:

- 1 Type **1** to display a list of statements from which you can select a parent statement.
- 2 Type **2** to display a panel on which you can specify the relationship that exists between fields.
3. If you selected Option **1**, the Statement List (ABX\$RRLS) panel opens to display a list of result set statements from which you can select one to be used.

```
ABX$RRLS  V1R1 ----- Statement List ----- 2005/01/06 14:13:44
Option  ===>                               Scroll ===> PAGE

Session Name
Record Type Name

Cmd Resultset Name

Valid Line Commands: S - Select
```

Figure 74. Statement List (ABX\$RRLS) panel

4. If you selected option **2**, the Relation Field List (ABX\$RREL) panel is displayed. This panel permits you edit or delete fields that comprise a relationship between the statements identified in the Statement Name and Parent Statement Name fields. You can also choose to add a new set of relation fields to the list by entering the field names in the appropriate text entry fields of the Edit Relation Fields (ABX\$RREE) panel. On the Edit Relation Fields panel, you can press **Enter** in a text entry field to see a list of fields from which you can choose the field to be used.

Edit Query Criteria

1. If you specified **6** on the Option line of the Edit Query panel, IBM File Export displays the Query Criteria (ABX\$QCRT) panel. This panel displays a list any previously defined WHEN in COLUMN clauses and permits you to define additional clauses. These clauses are used to filter the intermediate result set of data extracted from DB2 by IBM File Export.
2. Complete the fields on this panel.

```

ABX$QCRT  V1R1 ----- Query Criteria ----- 2005/01/06 13:30:53
Option  ==>>                               Scroll ==>> PAGE

Session Name

Query Name

Row 1 of 1

Cmd  Value

Valid Line Commands  : D - Delete Line

Valid Option Commands: A - Add Line

```

Figure 75. Edit Query Criteria (ABX\$QCRT) panel

Panel fields:

Session Name

Displays the name of the session.

Query Name

Displays the name of the query.

Cmd Type any line command available on this panel in this column next to the desired entry.

Value

Specify a WHEN column IN clause to be used as a filter on the intermediate result set of data that DB2 has extracted.

Valid Line Commands:

D Type **D** to delete an entry from the list.

Valid Option Commands:

A Type **A** to add a line on which you can specify an additional WHEN column IN clause.

3. Add WHEN column IN clauses to or delete them from the displayed list.

Specify target table information

1. If you specified 7 on the Option line of the Edit Query panel, IBM File Export displays the Target Specification panel. The Session Name and Query Name are provided for you on that panel. Use this panel to provide information about the target table or file and the data to be written to it. You can create a target that is a DB2 table or a flat file.

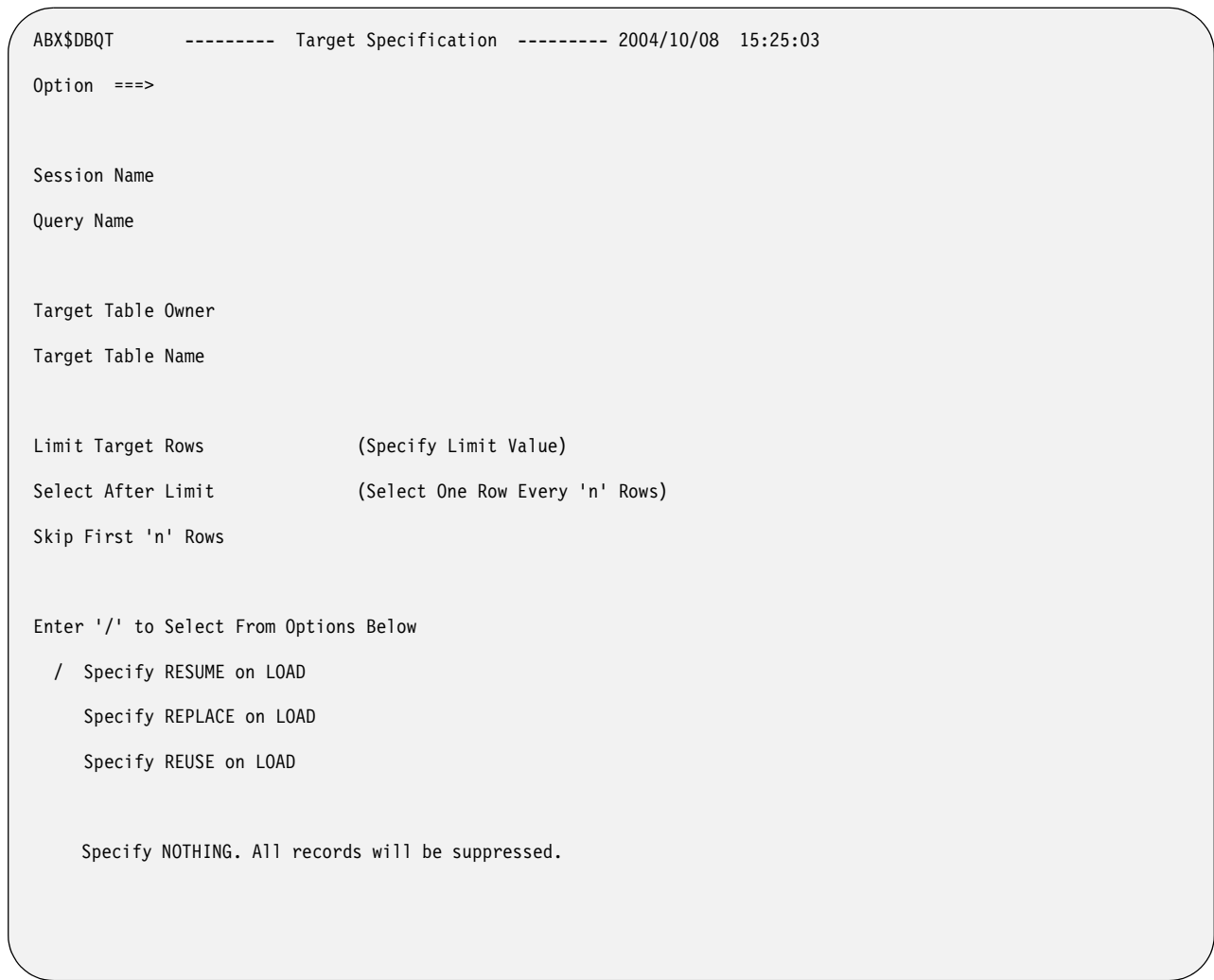


Figure 76. Target Specification (ABX\$DBQT) panel

Panel Fields:

Session Name

Specifies the name of your current session.

Query Name

Specifies the name under which the query is stored.

Target Table Owner

Type the name of the owner of the target table.

Target Table Name

Type the name you are assigning to the target table.

Limit Target Rows

Type the maximum number of rows to be written to the target.

Select after Limit

Type a value that indicates that, after the limit has been reached, every 'nn'th row thereafter will be written out to the target.

Skip First 'nn' Rows

Type a value that indicates that the first 'nn' rows selected to be written to the target are to be skipped before any rows are written to the target.

Specify RESUME/REPLACE/REUSE on LOAD

Resume, Replace, and Reuse are options for the DB2 LOAD Executor.

Specify NOTHING...

If you choose this option, IBM File Export will suppress writing the records that are produced by the SELECT statement.

2. To create a target table, specify the table owner and the table name in the appropriate fields.
3. Describe the output data using the next three fields.
 - Limit the number of rows written to the output table by typing a value here. The default is 0.
 - If you want to include an additional number of rows in your output table after the limit is reached, specify a value 'nn' here. After the limit is reached, then every 'nn'th row will also be written to the output table. The default is 0.
 - You can specify that you want to skip some number of selected rows before writing rows to the output table. The default is 0.
4. Specify the LOAD Executor option you want:
 - Specify RESUME on LOAD
Select this option to specify whether records are to be loaded into an empty or non-empty table space. Specifying RESUME continues loading into a non-empty table space.
 - Specify REPLACE on LOAD
Replace indicates whether the table space and its indexes need to be reset to empty before records are loaded.
 - Specify REUSE on LOAD
REUSE is used with REPLACE and specifies that LOAD should logically reset and reuse DB2-managed data sets without deleting and redefining them.
5. When you have completed work on this panel, press **PF3** to return to the Edit Query panel.

Specify intermediate data set DD statements

1. If you specified 8 on the Option line of the Edit Query panel, IBM File Export displays the DD Statement Definition (ABX\$DDE1) panel. Use this panel to provide DD information for the intermediate data set. The intermediate data set holds the extracted DB2 information from the source DB2 table before it is loaded into the target.
2. Complete the fields on this panel.

```

ABX$DDE1      ----- DD Statement Definition ----- 2004/06/18 10:48:44
Option  ==>                               Scroll ==> PAGE
DSN  :
ENTER mmmmm DD PARAMETERS:

Cmd  Text
DSN=,
DISP=SHR

Valid Line Commands  : D - Delete line
Valid Option Commands: A - Add line T - Generate statement from template
                       W - Generate DD statement with wizard

```

Figure 77. DD Statement Definition (ABX\$DDE1) panel

Panel fields:

DSN:
 Type a data set name. Type a partially qualified name with an asterisk (*) on the line to open ABX\$DSLS, which displays a list of data set names from which you can choose the one you want to work with.

Valid Line Commands:

D Type **D** in the Cmd field delete a DD parameter.

Valid Option Commands:

A Type **A** on the Option line to add a line on which you can type another DD parameter.

T Type **T** on the Option line to use an existing data set definition as a template for your DD parameters.

W Type **W** on the Option line to use a wizard to create your DD parameters.

3. On this panel, you can add or delete DD statements, select a template to use to create DD statements, or use the Data Set Allocation Parameters Wizard (ABX\$DDWZ).

```

ABX$DDWZ      ---- Data Set Allocation Parameters ---- 2004/10/04 13:27
Option  ==>>

Data Set unit          (If SMS managed, leave blank)

Volume serial          (If SMS managed, or unit specified
                        a tape unit, leave blank)

Space units            (CYL or TRK)

Primary quantity

Secondary quantity

Press ENTER to process, F12 to cancel

```

Figure 78. Data Set Allocation Parameters (ABX\$DDWZ) panel

Panel fields:

Data Set Unit:

Type the name of the data set device that will be used. Leave this field blank if this is an SMS-managed data set.

Volume Serial

Type the volume serial in this field. Leave this field blank if this is an SMS-managed data set or if you have specified a value for Data Set Unit.

Space Unit

Type **CYL** to use cylinders or **TRK** to use Tracks.

Primary Quantity

Type the number of tracks or cylinders to be allocated as the primary quantity.

Secondary Quantity

Type the number of tracks or cylinders to be allocated as the secondary quantity.

Generate JCL

1. To generate JCL containing your exported data, on the DB2 Menu (ABX\$DB2W) panel, select **4, Generate JCL**. Panel ABX\$JCLG, the JCL Generation panel, opens.

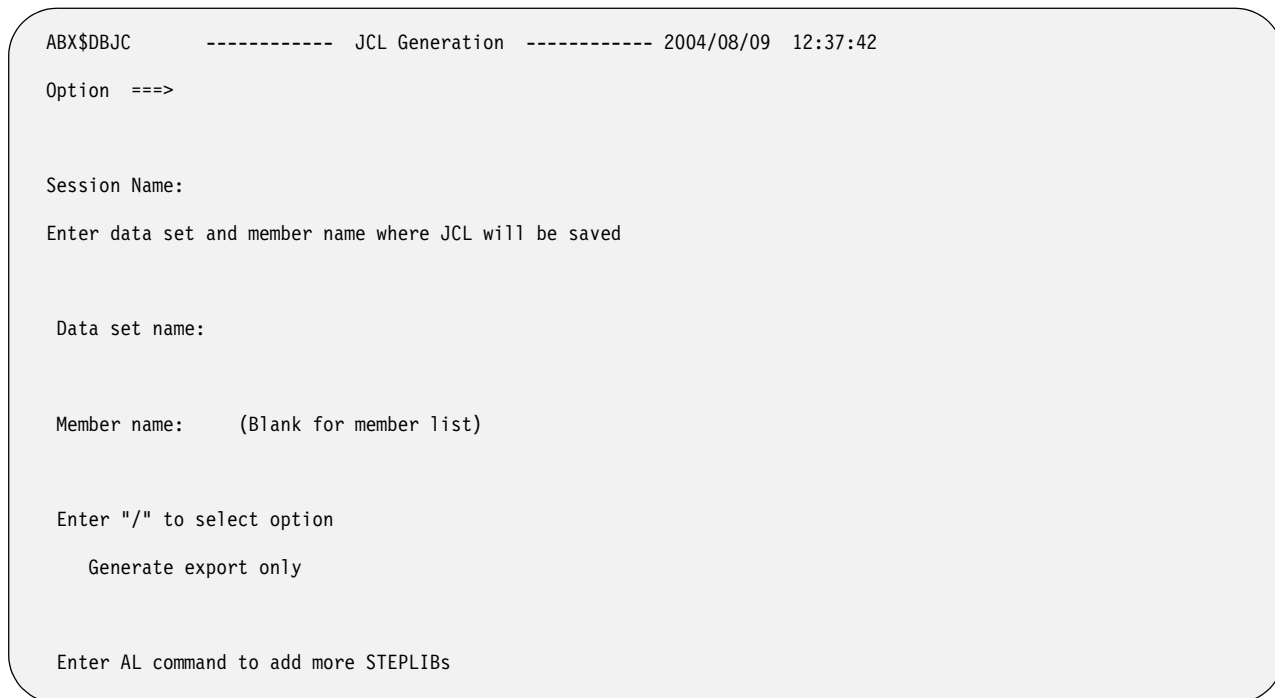


Figure 79. JCL Generation (ABX\$DBJC) panel

Panel fields:

Session Name

Lists the name of the IBM File Export for z/OS session.

Data set name:

If the data set name is not displayed when this panel opens, type the name of the data set in which the JCL will be saved. To browse for the data set name, enter a partial qualifier and an asterisk to display a list from which you can choose.

Member name:

If the member name is not displayed when this panel opens, type the name of the member in which the JCL will be saved.

Generate Export Only:

Type a slash in the field provided if you want to generate JCL for an export but not a LOAD.

Valid Option Commands:

AL Type **AL** on the Option line to add more STEPLIBs to your JCL definition.

2. On this panel, if the Data Set Name and Member Name are not displayed when the panel opens, type these names to indicate where you want to store the generated JCL. If you want to browse for the data set name, enter a partially qualified name and press **Enter**. IBM File Export for z/OS opens panel ABX\$DSL S which displays a list of data sets from which you can choose.
3. If you left the member name blank, panel ABX\$MEML displays a list of members from which you can choose the member to use by entering **S** in the Cmd field. After you have selected a member from this display, pressing **Enter** returns you to the JCL Generation panel.

```

ABX$MEML      ----- Member Selection List ----- 2004/06/21 13:16:55
Option ==>>>                               Scroll ==>> PAGE

                                                Row 1 of 23

Cmd Member Name
  $IDCAMS
  $ZAP
  ABX$SESS
  ABX123
  APFAUTH
  BACKABX
  COMPTST
  DFSVSAMP
  DSPURX00
  FIRSTFF
  ICCNTL
  IMSFF
  INITRCN

Valid Line Commands: S - Select

```

Figure 80. Member Selection List (ABX\$MEML) panel

Panel fields:

Cmd Type any valid line command in this field.

Member Name

Displays a list of members from which you can choose the member whose name will be filled in on the JCL Generation panel.

Valid Option Commands:

S Type **S** in the Cmd field to select this entry.

4. To add more STEPLIBs to your generated JCL, type **AL** on the Option line to display panel ABX\$DSSL.

```

ABX$DSSL      ----- IBM File Export for z/OS ----- 2004/06/21 14:25:36
Option ==>>>                               Scroll ==>> PAGE

Cmd Dataset

Valid Option Commands : A - Add Data Set
Valid Line Commands   : D - Delete Data Set

```

Figure 81. ABX\$DSSL panel

Panel fields:

Cmd Type any line command available on this panel in this column next to the desired entry.

Data set

Displays a list of data sets from which you can choose additional STEPLIBs.

Valid Option Commands:

A Type **A** on the Option line to add a data set to this panel..

Valid Line Commands:

D Type **D** on the Cmd line to delete a data set from the list.

5. There are two ways to export your data using IBM File Export for z/OS. You can run IBM File Export for z/OS, then load the data that it exported yourself, or you can run IBM File Export for z/OS and the DB2 Load Executor in sequence which results in your data being loaded into your DB2 target table when the process is complete.
6. Once you have completed work on panel ABX\$DSSL, press **PF3** to return to the JCL Generation (ABX\$DBJC) panel.
7. Pressing **PF3** again returns you to the DB2 Menu (ABX\$DB2W).

Chapter 8. Exporting IMS data

This chapter explains how to use IBM File Export for z/OS to export IMS data. The panels that are used in this process and their associated options, fields, and line commands are also explained. Note that the panels illustrated in this book are representative of the panels in the product and may differ slightly from the product panels.

IBM File Export provides for input from and output to IMS in these ways:

1. When accessing IMS using any of the supported IMS processing options, IBM File Export uses as input a sequential file produced by IBM's HD Reorganization Unload Utility (DFSURUL0) and produces a sequential output file that can be used by IBM's Reorganization Reload Utility (DFSURGL0). The product supports all database types that are supported by these utilities.
2. When accessing IMS using the DLI, DBB, or BMP processing options, you can access IMS databases directly without using unload files.
3. Using a control syntax that is similar to that of ABXUTIL, ABXFELD0 permits you to insert IMS segment data into IMS databases using the services of the DLI, DBB, or BMP processing mode. You can limit the databases that are to be processed by providing an INCLUDE and EXCLUDE mechanism. You can also read segment data from a database and write that data to a differently-named database. For more information, see "Inserting data into IMS databases using the File Export UNLOAD file" on page 259.

Fast Path Data Entry Databases (DEDB) are also supported when IBM's Fast Path Basic Tools (FPB) is installed and available for use.

Select the IMS source and target types

When you initiate a new IMS session from the IBM File Export Session Management panel, ABX\$SMNG (see page 102), select option **2, IMS**, to provide specifications for an IMS full-function or Fast Path session. You must specify a session name for the session you are creating. If this is a new session, the IMS Source and Target Types Specification (ABX\$IMST) panel opens.

```

ABX$IMST      ----- IMS Source and Target Type Specification -----
Option  ==>>

Session Name  MDSTITIW

Select the Source IMS type and the Target IMS type by typing a slash '/'
next to the IMS type.

Select the Source IMS Type:          Select the Target IMS Type:

IMS                                  IMS
DEDB                                 DEDB
DLI                                  DLI
DBB                                  DBB
BMP                                  BMP
                                     FEUNLOAD

```

Figure 82. IMS Source and Target Types (ABX\$IMST) panel

Based on the source and target pair you select, you will be shown the appropriate panels on which to define the source and target types. Valid source-target combinations are found in Table 1 on page 101. If you have specified an IMS or DLI source, a pop-up requests that you specify an IMS subsystem ID. Enter an ID or press **Enter** to see a list of available SSIDs. Press **PF3** to return to the IMS Source and Target Types panel. Pressing **PF3** again returns you to the IMS Menu (ABX\$IMSW).

Summary

The IMS Menu

“The IMS Menu” on page 135

Specify session settings

“Specify IMS session settings” on page 136

Specify source database

“Specify source database” on page 140

Specify target database

“Specify target database” on page 144

Specify variables and literals

“Define variables and literals” on page 148

Specify external relationships

“Define relationships” on page 150

Define segment specification and manipulation

“Define segment specification and manipulation” on page 154

Create DD statements

“Create DD statements” on page 166

Generate JCL

“Generate JCL” on page 170

The IMS Menu

From the IBM File Export Session Management panel, ABX\$SMNG (see page 102), select option **2, IMS**, to provide specifications for an IMS full-function or Fast Path session. You must specify a session name for the session you are creating.

Note: If you want to work with an existing session, specifying **4** on the Option line of the IBM File Export Session Management (ABX\$SMNG) panel together with a session name in the Session Name field opens the specified session. Specifying **5** on the Option line displays a list of previously-defined sessions from which you can choose the one to work with.

Use the IMS Menu (ABX\$IMSW) to provide IMS settings and to direct the balance of your IMS activities.

```
ABX$IMSW      -----  IMS Menu  -----  2004/10/07  15:53:56
Option  ==>

Session Name

Settings Specification
  0  Session Settings

Database specification
  1  Source Database --> Not selected
  2  Target Database --> Not selected

Data specification
  3  Variable and Literal Specification
  4  Relationship Specification
  5  Segment Specification and Manipulation

JCL specification
  6  Create DD Statements
  7  Generate JCL
```

Figure 83. IMS Menu (ABX\$IMSW) panel

Panel fields:

Session Name

Lists the name of the IBM File Export for z/OS session.

Valid Option | Commands:

- 0 Select this option to specify IMS session options. When you select this option, the IMS Session Settings (ABX\$IOPT) panel opens.
- 1 Select this option to define a source IMS database. When you select this option, a panel appropriate to the source processing option you selected opens.
- 2 Select this option to define a target IMS database. When you select this option, a panel appropriate to the target processing option you selected opens.
- 3 Select this option to create variables and literals to be used during this session. When you select this option, the Variables and Literals (ABX\$VLST) panel opens.
- 4 Select this option to create external relationships to be used during this session. When you select this option, the Relationship Specification (ABX\$RELS) panel opens.
- 5 Select this option to specify the source segments to be used and how the source segments should be manipulated to produce the target segments. When you select this option, IMS Segment Specification & Manipulation (ABX\$SEGM) panel opens.
- 6 Select this option to create DD statements. When you select this option, the DD Statements List (ABX\$IMSD) panel opens.
- 7 Select this option to generate JCL that is used to create target output. When you select this option, the JCL Generation (ABX\$FFJC) panel opens.

Specify IMS session settings

Use the IMS Settings (ABX\$IOPT) panel to specify settings specific to the current session. When specified on the IMS Settings panel, values will override values that have been specified in the IMS Options block. For expanded information about the keywords on this panel, see Appendix E, "IBM File Export IMS Options," on page 283.

ABX\$IOPT V1R1 ----- IMS Session Settings ----- 2004/11/04 11:45:13

Option ==>

Session Name

Wild Card Char PERCENT Decimal DECIMAL

SQL Output N (Yes / No) SQL Qualifier

File Export IMS processing Options:

SSID Name Get request PROCOPT

Use Dynamic PSBs N Dynamic PSB name prefix

Dynamic PSB name suffix 0 DLT Duplicate Segs

Alternate DFSRRC00 name

PSB Macro DSN

DBD library DSN

DOPT ACBLIB DSN

Reslib1 DSN

Reslib2 DSN

Reslib3 DSN

DFSVSAMP DSN/MBR

IMS Region Options:

AGN ALTID

BKO BUF

CPUTIME DBRC

DIRCA EXCPVR

FMT0 GSGNAME

IMSPLEX IRLM

IRLMNM LOCKMAX

MON NBA

OBA PARDLI

SSM STIMER

SWAP

Figure 84. IMS Session Settings (ABX\$IOPT) panel

Panel Fields:**Session Name**

Displays the name of the current session.

Wild Card Char

Type the character that you want to use as a wildcard in IBM File Export for z/OS in this field. The wild card is used in the LIKE predicate and represents the multi-character non-matching mask. Use this field to override global settings for this keyword.

Decimal

Type **Decimal** or **Comma** to specify the character to be used as the decimal point. Use this field to override global settings for this keyword.

SQL Output

Specify whether (Y) or not (N) you want to create SQL output.

SQL Qualifier

Type the name of the qualifier to be used by SQL for unqualified objects.

SSID Name

Specify the name of the IMS subsystem.

Get request PROCOPT

Specify the IMS processing options (PROCOPTs) that should be used to read databases when using a dynamically-built PSB. Valid options are **G** and **GO**. Valid for DLI, DBB, and BMP processing modes.

Use Dynamic PSBs

Specify whether (Y) or not (N) the Dynamic PSB feature should be used for this execution. Valid for DLI, DBB, and BMP processing modes. For more information about the use of static and dynamic PSBs, see "Understanding static and dynamic PSBs" on page 290.

Dynamic PSB name prefix

Specify the 1- to 6-character prefix to be used for the names of the PSBs that are generated by the Dynamic PSB feature. This prefix, together with a 2-character suffix, comprises the PSB's name. Valid for DLI, DBB, and BMP processing modes.

Dynamic PSB name suffix

Specify the 2-character numeric suffix to be used for the names of the PSBs that are generated by the Dynamic PSB feature. This suffix, together with the prefix you specified, comprises the PSB's name. Valid for DLI, DBB, and BMP processing modes.

DLT Duplicate Segs

Specify whether (Y) or not (N) IBM File Export should automatically delete any duplicate segments that are found in an IMS database when writing to that database. Valid for DLI, DBB, and BMP processing modes.

Alternate DFSRRC00 name

Specify a 1- to 8-character program name that allows you to execute an alternate IMS region controller program to DFSRRC00. Valid for DLI, DBB, and BMP processing modes.

PSB Macro DSN

Specify a 1- to 44-character name of the partitioned data set that contains the IMS macros used to generate PSBs when using the Dynamic PSB feature. Valid for DLI, DBB, and BMP processing modes.

DBD Library DSN

Specify a 1- to 44-character name of the partitioned data set that contains the DBDLIB members associated with the IMS subsystem. Valid for DLI, DBB, and BMP processing modes.

DOPT ACBLIB DSN

Specify a 1- to 44-character name of the partitioned data set that contains the ACBLIB members that are created by IBM File Export when the Dynamic PSB feature is used while processing. Valid for BMP processing mode.

Reslib1 DSN

Specify a 1- to 44-character name of the SVC data set that contains the IMS nucleus for the IMS subsystem. Valid for DLI, DBB, and BMP processing modes using the Dynamic PSB feature.

Reslib2 DSN

Specify a 1- to 44-character name of the data set that contains any IMS dynamic allocation members for the IMS subsystem. Valid for DLI, DBB, and BMP processing modes using the Dynamic PSB feature.

Reslib3 DSN

Specify a 1- to 44-character name of the data set that contains any user programs or exits needed by IMS to access IMS databases. Valid for DLI, DBB, and BMP processing modes using the Dynamic PSB feature.

DFSVSAMP DSN

Specify a 1- to 44-character name of the data set that contains the buffer and subpool information for databases. Valid for DLI, DBB, and BMP processing modes.

AGN Specify the Application Group Name used for interregion communication security. Valid for BMP processing mode.

ALTID

Specify an alternate IMSID that a BMP region will attempt to connect to if the primary subsystem ID (IMSID) is not available for processing. Valid for BMP processing mode.

BKO Specify whether (Y) or not (N) dynamic backout should occur if a pseudo-abend occurs. Valid for DLI and DBB processing modes.

BUF Specify a 1- to 3-digit number that identifies the number of 1K blocks to be used in calculating the size of the OSAM buffer subpools. Valid value is a number from 000 through 999. Valid for DLI and DBB processing modes.

CPUTIME

Specify 1 to 4 numeric digits which identify the maximum task time in minutes that a BMP region may consume. Valid for BMP processing mode.

DBRC

Specify whether (Y) or not (N) DBRC should be used for this execution. Leaving this parameter blank indicates that the DBRC= value specified on the IMSCTRL macro during IMS system definition should be used. Valid for DLI and DBB processing modes.

DIRCA

Specify 3 numeric digits that identify the number of 1K blocks of storage to be used as the interregion communications area. Valid values are 0 through 999.

EXCPVR

Specify whether (Y) or not (N) OSAM buffer pools should be page-fixed. Valid for DLI and DBB processing modes.

FMT0

Specify a 1-character value to identify the format of any IMS dumps. Valid for DLI and DBB processing modes. Because many values are supported, please refer to *IBM IMS Installation Volume 2: System Definition and Tailoring*, for information about the supported values.

GSGNAME

Specify a 1- to 8-character Global Services Group name that should be used if IRLM is active for this execution. This parameter is valid for DLI and DBB processing modes.

IMSPLEX

Specify the 5-character name that is passed to the SCI registration exit, DSPSCIX0. Valid for DLI and DBB processing modes.

IRLM

Specify whether (Y) or not (N) IRLM should be used for this execution. Leaving this parameter blank indicates that the IRLM= value specified on the IMSCTRL macro during IMS system definition should be used. Valid for DLI and DBB processing modes.

IRLMNAME

Specify a 1- to 8-character name for the IRLM subsystem that should be used if IRLM is active for this execution. This parameter is valid for DLI and DBB processing modes.

LOCKMAX

Specify a 1- to 5-digit numeric value that may be used to override the LOCKMAX value that was specified in a PSBGEN. Valid for DLI, DBB, and BMP processing modes.

MON Specify whether (Y) or not (N) the IMS monitor should be active for this execution. Valid for DLI and DBB processing modes.

NBA Specify 1 to 4 numeric digits that identify the number of IMS buffers that should be used when processing Fast Path databases should the number of buffers specified prove to be insufficient. Valid values are 0 through 9999. Valid for BMP processing mode.

OBA Specify 1 to 4 numeric digits that identify the number of IMS buffers that should be used as overflow when processing Fast Path databases should the number of buffers specified prove to be insufficient. Valid values are 0 through 9999. Valid for BMP processing mode.

PARDLI

Specify **1** in this field to indicate that DL/I access to the IMS databases should occur in the IMS control region. Specify **0** in this field to indicate that DL/I access to the IMS database should occur in the BMP region.

SSM Specify the suffix of a member name used to control access to external subsystems. Valid for BMP processing mode.

STIMER

Specify whether you want to gather processor time statistics. Refer to the IBM *IMS Installation Volume 2: System Definition and Tailoring* manual for information about this parameter (STIMER) and the meaning of various parameter values you may specify.

SWAP

Specify whether (Y) or not (N) the address space is swappable. Valid for DLI and DBB processing modes.

Valid Option Commands:

S Specify to create SQL output DD.

c Specify to connect to your default repository.

Specify source database

Follow these steps to define your source database.

If your source processing option is IMS or DLI ...

1. On the IMS Menu panel, select option **1, Source Database**. The IMS Source Database Specification panel, ABX\$FFDB, opens.
2. On this panel, the name of your session is displayed in the **Session Name** field. Use this panel to define your source database.

```
ABX$FFDB      ----- IMS Source Database Selection -----  
Option  ==>  
Session Name  
  
Full-Function DB:  
  DBD Library  
  
Database      (Blank for member list)
```

Figure 85. IMS Source Database Selection (ABX\$FFDB) panel

Panel fields:

Session Name

Lists the name of the IBM File Export for z/OS session.

DBD Library

Lists the name of the source IMS DBD library that contains the DBD for HIDAM/HDAM/HISAM databases.

Database

Type the name of the source database. If you would rather select from a list, leave this field blank. When you press **Enter**, IBM File Export displays a list of databases (ABX\$DSLS) from which you can choose the one you want to work with.

```
ABX$DSLS      ----- IBM File Export for z/OS ----- 2004/10/20 15:02:31  
Option  ==>                               Scroll ==> PAGE  
  
Select source database library:  
  
Cmd  Data Set Name                          Row 1 of nnn  
  
Valid Line Commands:  S - Select
```

Figure 86. ABX\$DSLS panel

Panel fields:

Cmd Type any valid line command in the Cmd field to select the associated object.

Valid Line Command:

S Type **S** in the Cmd field to select the DBDLIB you want to work with.

3. Once you have chosen the DBD library and database you want to work with from the list, press **Enter** to return to the IMS Source Database Specification panel.

IBM File Export has filled in the Database field on this panel with your selection from the selection list. Pressing **PF3** returns you to the IMS Menu.

If your source processing option is DEDB...

1. On the IMS Menu (ABX\$IMSW) panel, select option **1, Source Database**. The IMS Source Database Specification panel, ABX\$FPSR, opens.
2. On this panel, the name of your session is displayed in the **Session Name** field. Use this panel to define your source database. Press **PF3** to return to the IMS Menu.

```

ABX$FPSR      ----- IMS Source Database Specification -----
Option  ==>>

Session Name:

DEDB Interface

DBDFN Library
  
```

Figure 87. IMS Source Database Specification (ABX\$FPSR) panel

Panel fields:

Session Name

Lists the name of the IBM File Export for z/OS session.

Fast Path (DEDB) DBDFN Library

Type the name of the Fast Path Data Entry Data Base DBDFN library. If you would rather select from a list, you can specify a high-level qualifier in the field. When you press **Enter**, IBM File Export displays a list of data sets (ABX\$DSLS) from which you can choose the one you want to work with. To browse for a name, you must enter at least a high-level qualifier in the DBDFN Library field, together with an asterisk, or no further action is taken.

```

ABX$DSLS      ----- IBM File Export for z/OS ----- 2004/06/17 16:56:18
Option  ==>>                               Scroll ==>> PAGE
Select DBDFN Library:

Cmd  Data Set Name

Valid Line Commands: S - Select
  
```

Figure 88. ABX\$DSLS panel

Panel fields:

Session Name

The name of the current session.

Cmd Type any line command available on this panel in this column next to the desired object.

Data Set Name

Lists the data set names.

Valid Line Commands:

S Type S in the Cmd field to select a DBDFN library.

3. Once you have chosen the data set you want to work with from the list, press **PF3** to return to the IMS Source Database Specification panel. IBM File Export has filled in the DBDFN Library field on this panel with your selection from the Dataset Selection List. Pressing **PF3** again returns you to the IMS Menu.

If your source processing option is DBB or BMP...

1. On the IMS Menu (ABX\$IMSW) panel, select option 1, **Source Database**. The IMS Source Database Specification panel, ABX\$FPTG, opens.
2. On this panel, the name of your session is displayed in the **Session Name** field. Use this panel to define your source database.

```
ABX$FPTG      ----- IMS Source Database Specification -----  
  
Option ==>  
  
  Session Name:  
  
Fast Path (DEDB)  
  
  Do you want to use ACBLIB or MODSTAT?  (A or M)  
  
ACBLIB  
MODSTAT  
IMSACBA  
IMSACBB  
  
DATABASE      (Blank for member list)
```

Figure 89. IMS Source Database Specification (ABX\$FPTG) panel

Panel fields:

Session Name

Lists the name of the IBM File Export for z/OS session.

Do you want to use ACBLIB or MODSTAT?

Type **A** on this line to use ACBLIB or **M** to use MODSTAT. .

ACBLIB

Type the name of the IMS ACBLIB data set in this field. This field is mandatory unless you choose to specify the MODSTAT, IMSACBA, and IMSACBB fields. In that case, leave this ACBLIB field blank. You can browse for this name by entering a partially qualified name together with an asterisk to display a list from which you can select the data set you want to work with.

MODSTAT

Type the name of the target IMS MODSTAT data set in this field. Leave this field blank if you fill in the ACBLIB field. If you fill in this MODSTAT field, you must also specify IMSACBA and IMSACBB fields. You can browse for this name by entering a partially qualified name together with

an asterisk to display a list from which you can select the data set you want to work with.

IMSACBA

Type the name of the target IMSACBA data set in this field. Leave this field blank if you fill in the ACBLIB field. If you fill in this IMSACBA field, you must also specify MODSTAT and IMSACBB fields. You can browse for this name by entering a partially qualified name together with an asterisk to display a list from which you can select the data set you want to work with.

IMSACBB

Type the name of the target IMSACBB data set in this field. Leave this field blank if you fill in the ACBLIB field. If you fill in this IMSACBA field, you must also specify MODSTAT and IMSACBA fields. You can browse for this name by entering a partially qualified name together with an asterisk to display a list from which you can select the data set you want to work with.

Database

Type the name of the target Fast Path database. If you would rather select from a list, leave this field blank. When you press **Enter**, IBM File Export displays a list of databases (ABX\$DSLS) from which you can choose the one you want to work with.

```
ABX$DSLS      ----- IBM File Export for z/OS ----- 2004/06/23 13:25:30
Option  =====>                               Scroll =====> PAGE
                                                    Row 1 of 2

Cmd Database
      FPTST04
      FPTST05
```

Figure 90. ABX\$DSLS panel

Panel fields:

Cmd Type any line command available on this panel in this column next to the desired object.

Database

Lists the database names.

3. Once you have chosen the database you want to work with from the list, press **PF3** to return to the IMS Source Database Specification panel. IBM File Export has filled in the Database field on this panel with your selection from the Database Selection List. Pressing **PF3** again returns you to the IMS Menu.

Specify target database

Follow these steps to define your target database.

If your target processing option is IMS or DLI...

1. On the IMS Menu panel, select option **2, Target Database**. The IMS Target Database Selection panel, ABX\$FFDB, opens.
2. On this panel, the name of your session is displayed in the **Session Name** field. Use this panel to define your target database.

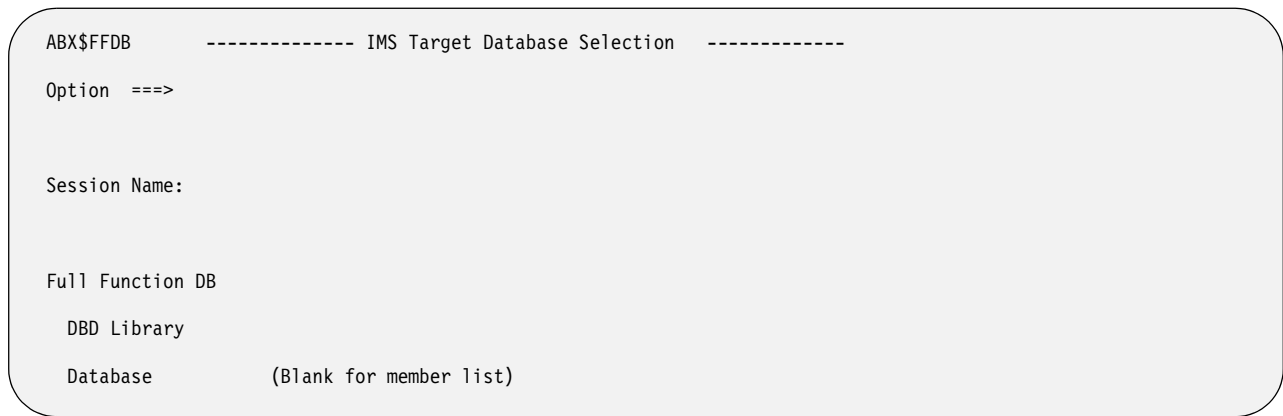


Figure 91. IMS Target Database Selection (ABX\$FFDB) panel

Panel fields:

Session Name

Lists the name of the IBM File Export for z/OS session.

DBD Library

Type the name of the target IMS DBD library that contains the DBD for HIDAM/HDAM/HISAM databases. You can enter a partial qualified name together with an asterisk to display a list from which you can select a library. If not specified, this field defaults to the source DBDLIB data set name.

Database

Type the name of the full-function database. If you would rather select from a list, leave this field blank. When you press **Enter**, IBM File Export displays a list of databases (ABX\$DSLS) from which you can choose the one you want to work with.

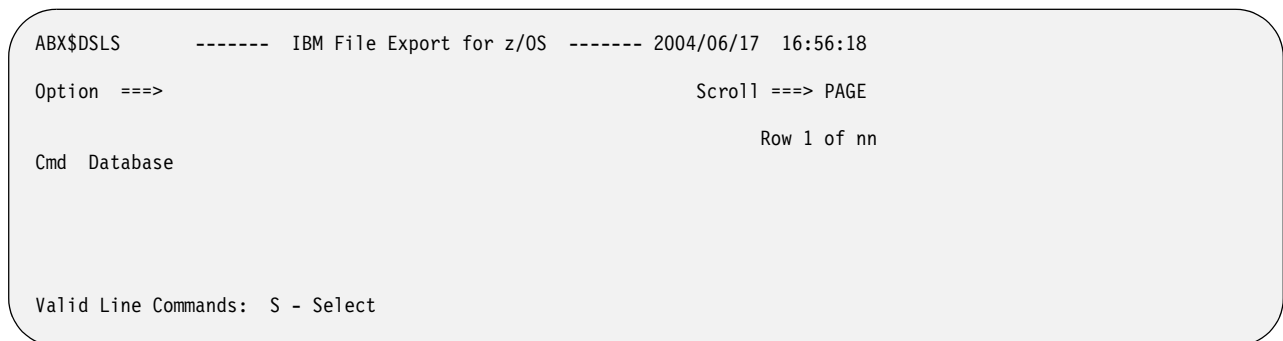


Figure 92. ABX\$DSLS panel

Panel fields:

Cmd Type any valid line command in the Cmd field to select the associated object.

Valid Line Command:

S Type **S** in the Cmd field to select the database you want to work with.

- Once you have chosen the database you want to work with from the list, press **Enter** to return to the IMS Target Database Specification panel. IBM File Export has filled in the Database field on this panel with your selection from the Database Selection List. Pressing **PF3** returns you to the IMS Menu.

If your target processing option is DEDB, DBB, or BMP...

1. On the IMS Menu panel, select option 2, **Target Database**. The IMS Target Database Specification panel, ABX\$FPTG, opens.
2. On this panel, the name of your session is displayed in the **Session Name** field. Use this panel to define your target database.

Note: If your source processing option was DBB or BMP, the appropriate fields on this panel are display fields, with the values in them taken from the IMS Source Database Specification panel. The exception is the Database field, where you can enter a database name or browse for the name.

```
ABX$FPTG      ----- IMS Target Database Specification -----
Option  ===>

  Session Name:

DEDB Interface

  Do you want to use ACBLIB or MODSTAT?  (A or M)

ACBLIB
MODSTAT
IMSACBA
IMSACBB

DATABASE      (Blank for member list)
```

Figure 93. IMS Target Database Specification (ABX\$FPTG) panel

Panel fields:

Session Name

Lists the name of the IBM File Export for z/OS session.

Do you want to use ACBLIB or MODSTAT?

Type **A** on this line to use ACBLIB or **M** to use MODSTAT. .

ACBLIB

Type the name of the target IMS ACBLIB data set in this field. This field is mandatory for DEDB databases unless you choose to specify the MODSTAT, IMSACBA, and IMSACBB fields. In that case, leave this ACBLIB field blank. You can browse for this name by entering a partially qualified name together with an asterisk to display a list from which you can select the data set you want to work with.

MODSTAT

Type the name of the target IMS MODSTAT data set in this field. Leave this field blank if you fill in the ACBLIB field. If you fill in this MODSTAT field, you must also specify IMSACBA and IMSACBB fields. You can browse for this name by entering a partially qualified name together with an asterisk to display a list from which you can select the data set you want to work with.

IMSACBA

Type the name of the target IMSACBA data set in this field. Leave this field blank if you fill in the ACBLIB field. If you fill in this IMSACBA

field, you must also specify MODSTAT and IMSACBB fields. You can browse for this name by entering a partially qualified name together with an asterisk to display a list from which you can select the data set you want to work with.

IMSACBB

Type the name of the target IMSACBB data set in this field. Leave this field blank if you fill in the ACBLIB field. If you fill in this IMSACBA field, you must also specify MODSTAT and IMSACBA fields. You can browse for this name by entering a partially qualified name together with an asterisk to display a list from which you can select the data set you want to work with.

Database

Type the name of the target Fast Path database. If you would rather select from a list, leave this field blank. When you press **Enter**, IBM File Export displays a list of databases (ABX\$DSLS) from which you can choose the one you want to work with.

```
ABX$DSLS      -----  IBM File Export for z/OS  -----  2004/06/23  13:25:30
Option  ==>>                               Scroll ==>>  PAGE
                                                Row 1 of 2

Cmd  Database
    FPTST04
    FPTST05
```

Figure 94. ABX\$DSLS panel

Panel fields:

Cmd Type any line command available on this panel in this column next to the desired object.

Database

Lists the database names.

3. Once you have chosen the database you want to work with from the list, press **PF3** to return to the IMS Target Database Specification panel. IBM File Export has filled in the Database field on this panel with your selection from the Database Selection List. Pressing **PF3** again returns you to the IMS Menu.

If your target processing option is FEUNLOAD

1. On the IMS Menu panel, select option **2, Target Database**. The IMS Target Database Specification panel, ABX\$FETG, opens.
2. On this panel, the name of your session is displayed in the **Session Name** field. Use this panel to provide a name for your FEUNLOAD data set.

```

ABX$FETG      ----- IMS Target Database Specification -----
Option  ==>

Session Name

FEUNLOAD:
  Data Set Name

```

Figure 95. IMS Target Database Specification (ABX\$FPTG) panel

Panel fields:

Session Name

Lists the name of the IBM File Export for z/OS session.

FEUNLOAD Data Set Name

Type the name of the target FEUNLOAD data set.

3. Press **PF3** to return to the IMS Menu.

Define variables and literals

Follow these steps to define or edit variables and literals. Literals are constants and can be either implicit or named. Variables are literals that change value during the process due to being incremented as a result of system actions.

1. On the IMS Menu panel, select option **3, Variable and Literal Specification**. Panel ABX\$VLST, containing a list of already-defined variables and literals, opens. On this panel, the name of your session is displayed in the **Session Name** field. Any variables or literals you have previously defined are displayed.
2. To define a variable, type **AV** on the Option line. To define a literal, type **AL** on the Option line.

```

ABX$VLST      ----- Variables and Literals ----- 2004/06/18 09:25:00
Option  ==>                                     Scroll ==> PAGE

Session Name

Row 1 of 2

Cmd  Label                Type
-----
TESTVAR                    L
TESTLIT                    V

Valid Line Commands : E - Edit Entry D - Delete Entry
Valid Option Commands: AL - Add Literal AV - Add Variable

```

Figure 96. Variables and Literals (ABX\$VLST) panel

Panel fields:

Session Name

Lists the name of the IBM File Export for z/OS session.

Cmd Type any valid line command in the Cmd field to select the associated object.

Label

Displays the name you have assigned to this variable or literal.

Type Displays **L** for Literal or **V** for Variable.

Valid Line Commands:

E Type **E** on a line to edit the associated literal or variable.

D Type **D** on a line to delete the associated literal or variable.

Valid Option Commands:

AL Type **AL** on the Option line to define a literal.

AV Type **AV** on the Option line to define a variable.

3. If you have entered **AL** or **AV** on the Option line, the New Literal or New Variable panel (ABX\$VARS) respectively, displays. On this panel, you can define a new variable or literal or edit an existing variable or literal.

```
ABX$VARS      ----- New Literal ----- 2004/06/18 09:41:42

Source Name

Length      0

Data Type   CHAR

Value

Press ENTER or F3 to process, F12 to cancel
```

Figure 97. New Literal (ABX\$VARS) panel

Panel fields:

Source Name

Type a name for the source data.

Length

Type the length of the source data in this field. Length is a numeric literal and is the length in bytes required in memory for the source data.

Data Type

Type the data type in this field. (For a description of these data types, see Table 3 on page 257.) The value can be one of the following:

- INT or INTEGER
- CHAR
- FLOAT
- HEX

- SHORT
- DOUBLE
- PACKED

Value

Type the value to be assigned to the variable or literal.

4. When you have completed work on this panel, press **PF3** to return to the Variables and Literals List. Pressing **PF3** again returns you to the IMS Menu.

Define relationships

Follow these steps to provide information to define external relationships. Before doing so, you may want to review the information about application-defined relationships in “Defining relationships in IBM File Export” on page 5.

1. On the IMS Menu panel, select option 4, **Relationship Specification**. Panel ABX\$RELS opens.

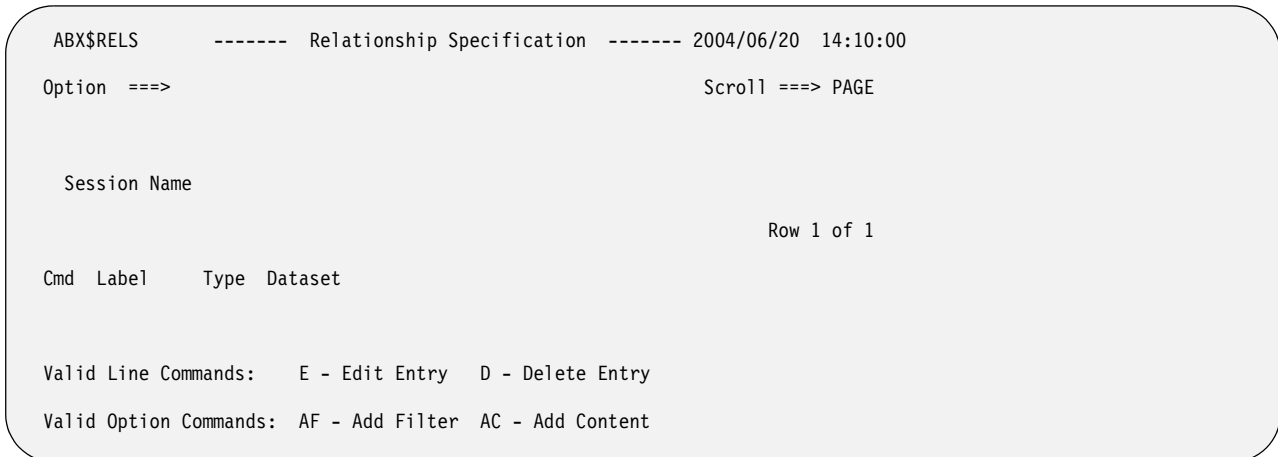


Figure 98. Relationship Specification (ABX\$RELS) panel

Panel fields:

Session Name

Lists the name of the IBM File Export for z/OS session under which you are working.

Panel columns:

Cmd Type any line command available on this panel in this column next to the desired entry.

Label

Lists the name you have assigned to this filter or content specification.

Type

Lists an **F** if this is a filter or **C** if this is a content specification.

Data Set

Names the data set with which filter or content relationship data has been associated.

Valid Line Commands:

E Use to edit an entry in this list. When you type the **E** line command in the **Cmd** field, the ABX\$RELF (if this line is for a filter) or ABX\$RELC (if this line is for a content specification) panel opens.

D Use to delete an entry from this list.

Valid Option Commands:

- AF** Use to add a filter to the current data set. When you type the **AF** option, the **ABX\$RELF** panel opens.
- AC** Use to add a content specification to the current data set. When you type the **AC** option, the **ABX\$RELC** panel opens.

2. If you entered **AF** to add a filter, the Filter Specification panel (**ABX\$RELF**) opens. Use this panel to add a filter. The name of the source data set that contains the related data that you want to add to your target is displayed in the Source Name field if you have selected that data set from the External Data Relationships list. Default values appear in the rest of the fields. You can change the value in any field by overtyping it.

Note: External data is that data contained in a data set from which source data will be selected to be exported to the target, as, for example, a VSAM file you want to export into your target table, or a sequential file from which you want to pull fields to be added to your target object.

```
ABX$RELF ----- Filter Specification ----- 2004/06/20 14:39:58

Source Name
DSN
Position      0
Length        0
Data Type     HEX

Press ENTER or F3 to process, F12 to cancel
```

Figure 99. Filter Specification (**ABX\$RELF**) panel

Panel fields:

Source Name

Type the name you are assigning to this filter.

DSN Type the name of the data set that is the source for this filter. If you put an asterisk in this field, panel **ABX\$DSL** opens. On this panel, you can select the data set you want to use.

Position

Type the position of the source data in this field. Position can be specified:

- As a numeric integer type constant like 1 or 100
- Relative to the end of the record like 'end - 10' (ten bytes from the end if the record).
- Using a hex constant (x'22' or h'22') which will always be interpreted as an offset (never as a 'position').

Length

Type the length of the content key or content data (as appropriate) in this field. Length is a numeric literal and is the length of the bytes required in memory for the source data.

Data Type

Type the data type of the source data set in this field. (For a description of these data types, see Table 3 on page 257.) You can specify any of the following types:

- INT
- CHAR
- FLOAT
- HEX
- SHORT
- DOUBLE
- PACKED

Pressing **PF3** returns you to the Relationship Specification panel, ABX\$RELS.

3. If you entered AC to add a content specification, the Relation Content panel (ABX\$RELC) opens.

```
ABX$RELC      ----- Relation Content ----- 2004/06/20 14:49:23

Content Name
DSN

Key For Content
  Position      0
  Length        0
  Data Type     HEX

Data For Content
  Position      0
  Length        0
  Data Type     HEX

Press ENTER or F3 to process, F12 to cancel
```

Figure 100. Relation Content (ABX\$RELC) panel

Panel fields:

Source Name

Type the name you are assigning to this content specification.

- DSN** Type the name of the data set that is the source for this content specification. If you put an asterisk in this field, panel ABX\$DSLS opens. On this panel, you can select the data set you want to use.

Position

Type the position of the source data in this field. Position can be specified:

- As a numeric integer type constant like 1 or 100
- Relative to the end of the record like 'end - 10' (ten bytes from the end of the record).
- Using a hex constant (x'22' or h'22') which will always be interpreted as an offset (never as a 'position').

Length

Type the length of the content key or content data (as appropriate) in this field. Length is a numeric literal and is the length of the bytes required in memory for the source data.

Data Type

Type the data type of the source data in this field. (For a description of these data types, see Table 3 on page 257.) You can specify any of the following types:

- INT
- CHAR
- FLOAT
- HEX
- SHORT
- DOUBLE
- PACKED

Note: The position, length, and data type fields are repeated first for the key and second for the content specification.

Use this panel to specify information about the content data set. The DSN field is filled in for you if you have selected the data set from the External Data Relationships list. Default values appear in the rest of the fields. You can change the value in any field by overtyping it.

To add content, you select data from a data set and match that data to the data from a specific column in the source in order to populate your target. Use the key to identify the key column. Preferably, that column is unique, as, for example, Account Number.

Use the data to identify actual data that is to be selected from the source to be written to the target.

To summarize, use a content specification to include data from an external source. Use the Key to specify what data IBM File Export is to compare to the source file, and the data to specify what data IBM File Export is to include in the target output when the key is found.

For example, assume that your source file contains addresses using two-character State abbreviations. But you would rather have addresses with the State name spelled out in the target data. To accomplish this, you would define a Content Relationship in File Export to use an external file that contains the abbreviated State name in columns 1 and 2, and the spelled out State name in columns 3 through 28. In this example, the Key Column in the external file would be the abbreviated State name (since the same data exists in the source file) in columns 1 and 2, and the Data Column that you want to include in the target data would be the data contained in columns 3 through 28. So that when File Export finds 'TX' in the source file as the abbreviated State name, for example, the Key from the external file is matched, and File Export uses the spelled-out State name from the external file to populate the target data file.

Note: External data is that data contained in a data set from which source data will be selected to be exported to the target, as, for example, a VSAM file you want to export into your target table, or a sequential file from which you want to pull fields to be added to your target object.

Pressing **PF3** returns you to the Relationship Specification panel, (ABX\$RELS).

4. Entering **E** on the Cmd line next to a filter or content specification, the ABX\$RELF (if this line is for a filter) or ABX\$RELC (if this line is for a content specification) panel opens. When you have completed your work on this panel, pressing **PF3** returns you to the Relationship Specification panel (ABX\$RELS).
5. Pressing **PF3** again returns you to the IMS Menu (ABX\$IMSW).

Define segment specification and manipulation

Follow these steps to provide the requisite information for segment specification and manipulation.

1. On the IMS Menu panel, select option **5, Segment Specification and Manipulation**. The IMS Segments (ABX\$SEGM) panel opens.

```

ABX$SEGM      ----- Segments ----- 2004/06/18  11:20:46
Option  ==>>                               Scroll ==> PAGE
      Session Name
      Database
                                           Row 1 of n
Cmd  Segment  Level  Skip    Limit    Then    Last
    ROOT      1
    L01H02    2
    L01H03    3
Valid Line Commands:  E - Edit Fields  T - Edit Types and Generate SELECT
  
```

Figure 101. Segments (ABX\$SEGM) panel

Panel fields:

Session Name

Lists the name of the IBM File Export for z/OS session under which you are working.

Database

Lists the name of the source database.

Panel columns:

Cmd Type any line command available on this panel in this column next to the desired segment.

Segment

Lists the segment name.

Level

Lists the segment hierarchical level.

Skip

Type the number of records that you want to skip. You can specify that you want to skip the first 'nn' accepted records from the source when you are creating your output.

Limit

Type the number of records that is the maximum number you will produce in the target output.

Then Type an integer in this field that specifies that after hitting the limit, 1 of the next 'n' acceptable records will be produced for the target.

Last Use this field to indicate whether this segment is the last segment. Type **N** in this field to indicate that this segment should not be the last segment included in the target output. Type **Y** to indicate that this segment is the last segment to be included in the target output.

Valid Line Commands:

E Use to edit a segment. When you type the **E** line command in the **Cmd** field of a segment, panel ABX\$FSEL opens.

T Use to edit the File Export SQL types (that is, for example, Debit or Credit) for the selected segment and generate the SELECT statement. When you type the **T** line command in the **Cmd** field of a segment, the Record Type List panel, ABX\$SOTL, opens.

Q Use to generate SQL output. Type **Q** on a Cmd line to add SQL output parameters using the SQL Output Parameters panel, ABX\$SPFI. On this panel, you can provide a table owner and table name, and specify the number of records to be written out. The **Q** command will only be available if you specified you wanted to be able to generate SQL output when defining your IMS settings.

2. Fill in or modify any fields on the Segments panel.
3. Use the **E** line command to add a field to the segment, delete a field from the segment, and reset the current fields for the segment as well as to modify the field name, start position, size, and type, and specify whether or not the field should be scrambled. When you type the **E** line command in the **Cmd** field of a segment, the ABX\$FSEL panel opens.
 - a. Use the Segment Fields (ABX\$FSEL) panel to modify parameters for user-defined fields. When the segment displayed is from the physical DBD, that is, it is not user defined, you cannot change the value of the parameters.

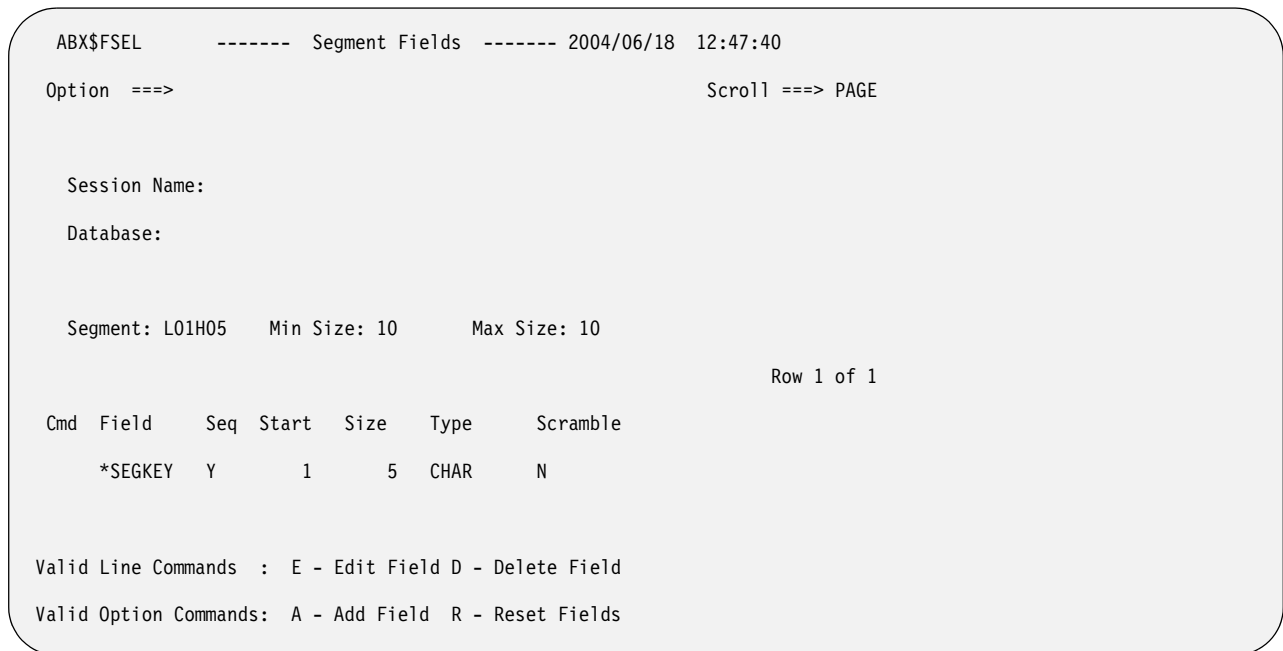


Figure 102. Segment Fields (ABX\$FSEL) panel

Panel fields:**Session Name**

Lists the name of the IBM File Export for z/OS session under which you are working.

Database

Lists the name of the source database.

Segment

Lists the name of the current segment.

Min Size

Lists the minimum size of the current segment.

Max Size

Lists the maximum size of the current segment.

Field Lists the current field name.

Seq Specifies whether or not this is a sequence field.

Start Lists the current start position.

Size Lists the current field size.

Type Lists the current data type, which may one of the following data types (see Table 3 on page 257) .

- INT or INTEGER
- CHAR
- FLOAT
- HEX
- SHORT
- DOUBLE
- PACKED

Scramble

Type **Y** in this field to indicate that this field should be scrambled. Leave the field blank or type **N** to indicate that the field should not be scrambled. You can change the value in this field by overtyping it. When a column is scrambled, the contents will retain the same format as in the source, but the contents will be randomized. This is useful when the source contains sensitive data. The algorithm used for scrambling produces reproducible output. This means that the output of a scrambled value will be the same each time that value is scrambled.

Valid Line Commands:

E Type **E** to edit a field from the segment

D Type **D** to delete a field from the segment

Valid Option Commands:

A Type **A** to add a field to the segment.

R Type **R** to reset the current segment to its default or previous values.

- b. On the Segment Fields (ABX\$FSEL) panel, when you type **A** to add a new field or **E** to edit an existing field, the Field Specification (ABX\$NFLD) panel opens. Use this panel to define new fields for the segment you are working with or edit an existing field.



Figure 103. Field Specification (ABX\$NFLD) Panel

Panel fields:

Field Name

Type a name for the field to be added to the segment.

Field Position

Type the position where the source data starts. Position can be specified:

- As a numeric integer type constant like 1 or 100
- Relative to the end of the record like 'end - 10' (ten bytes from the end if the record).
- Using a hex constant (x'22' or h'22') which will always be interpreted as an offset (never as a 'position').
- As a cumulative position/offset definition. For example:
 - Use the NEXT declarative to specify that the position is the next character after the prior field.

- Use the NEXTC declarative to specify that the position is the next character after the prior field.
- Use the NEXTI declarative to specify that the position is the next integer position after the prior field.
- Use the NEXTW declarative to specify that the position is the next word position after the prior field.
- Use the NEXTH declarative to specify that the position is the next halfword location after the prior field.
- Use the NEXTD declarative to specify that the position is the next double word location after the prior field.

Information: NEXTx does NOT set the field attribute (WORD/DOUBLE) just the position.

Field Length

Type the field length of the field.

Field Type

Type the data type for the field. (For a description of these data types, see Table 3 on page 257.) The data type may be one of the following:

- INT or INTEGER
- CHAR
- FLOAT
- HEX
- SHORT
- DOUBLE
- PACKED

Occurs

Type an integer that indicates the number of times a variable length repeating group occurs.

Depending On

Type the object of the Occurs. If the number of times a repeating group can occur in the record depends on this object of the Occurs, that dependency should be specified here. For information about Occurs Depending On, see “Understanding Occurs Depending On” on page 262.

Precision

Type the precision of the numeric data.

Scale Type the scale of the numeric field.

Signed

Specify whether or not the numeric data is signed.

Scramble

Type **Y** in this field to indicate that this field should be scrambled. Leave the field blank or type **N** to indicate that the field should not be scrambled. When a column is scrambled, the contents will retain the same format as in the source, but the contents will be randomized. This is useful when the source contains sensitive data. The algorithm used for scrambling produces reproducible output. This means that the output of a scrambled value will be the same each time that value is scrambled.

Group Length

Type the length of the group.

Maximum Length

Type the maximum length of the field.

Position Algorithm

Because the length of a group item containing variable length data is not a constant, an algorithm is required to express the position of a field, and in some cases, the length of the field or group item. Type the position algorithm in this field.

Group Length Algorithm

Because the length of a group item containing variable length data is not a constant, an algorithm is required to express the position of a field, and in some cases, the length of the field or group item. Type the group length algorithm in this field.

Enter "/" to select option

To specify an additional position algorithm, group length algorithm, or both, type a slash (/) in the appropriate field or fields.

- c. When you have completed defining the new field on this panel, press **Enter** to return to panel ABX\$FSEL, the Segment Fields panel. Press **PF3** to return to the Segment Selection and Manipulation panel (ABX\$SEGM).
4. Use the **T** line command to edit the record types for the selected segment and generate the SELECT statement. The Record Type List (ABX\$SQL) panel opens.

```
ABX$SQL      ----- Record Type List ----- 2004/08/04 10:31:27
Option ==>>                               Scroll ==>> PAGE
Session Name ABXIMS01
Row 1 of 2
Cmd Name          Excl Skip      Limit  Then
SIZE              N
DEFAULT           N
Valid Line Commands : E - Edit Type  D - Delete Type  F - Field List
                    S - SELECT statement list
Valid Option Command: A - Add New SQL Type  L - Load from Repository
```

Figure 104. Record Type List (ABX\$SQL) Panel

Panel fields:

Session Name

The name of the current session.

Panel columns:

Cmd Type any line command available on this panel in this column next to the desired segment.

Name Lists the SQL type name.

Excl If exclude is specified as **Y**, this type is excluded from processing.

Skip Type the number of records that you want to skip. You can specify that you want to skip the first 'nn' accepted records from the source when you are creating your output.

Limit Type the number of records that is the maximum number you will produce in the target output.

Then Type an integer in this field that specifies that after hitting the limit, 1 of the next 'n' acceptable records will be produced for the target.

Valid Line Commands:

- E** Type **E** to edit a type.
- D** Type **D** to delete a type.
- F** Type **F** to display a list of fields.
- S** Type **S** to display a list of SELECT statements.

Valid Option Commands:

- A** Type **A** to add a new record type.
- L** Type **L** to load data from the repository.
 - a. To delete a type, specify **D** in command field.
 - b. To add or edit a type, specifying **A** on the Option line or **E** in the Cmd field of a given type opens the Record Type (ABX\$SQLT) panel.

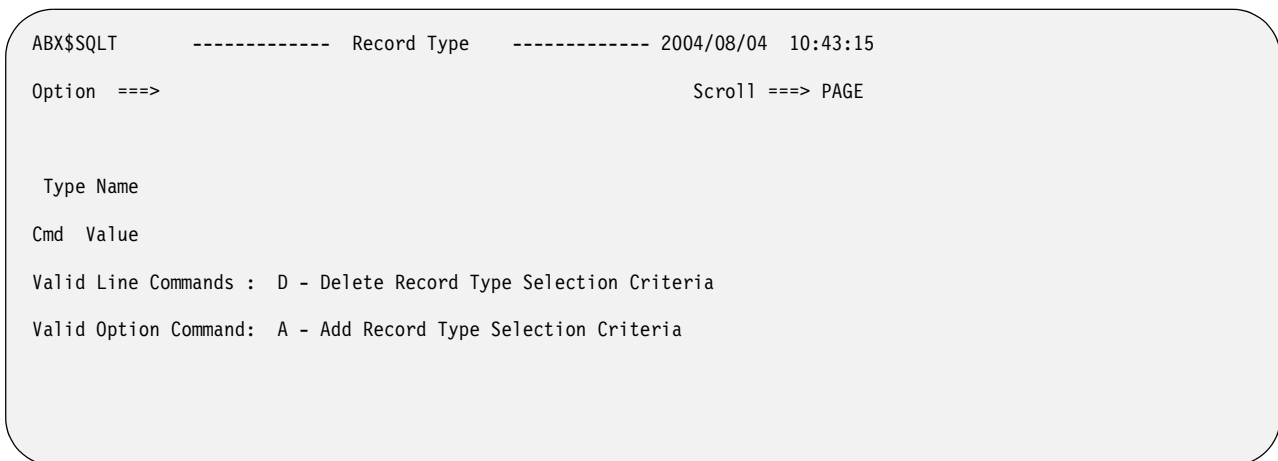


Figure 105. Record Type (ABX\$SQLT) Panel

Panel fields:

Type Name

Type a name (such as Debit or Credit) for the Record type to be added to the segment. If you are editing a type, the name will be supplied. You may edit the name by overtyping it.

Cmd Type any line command available on this panel in this column next to the desired segment.

Value

Type a WHEN and/or a WHERE predicate. Information valid for the record type is described in the control statement "TYPE" on page 251.

Valid Line Commands:

D Type **D** to delete a selection criterion.

Valid Option Commands:

A Type **A** to add a new selection criterion.

- 1) On the Record Type (ABX\$SQLT) panel, you can add or delete selection criteria for the type you have added or are editing. Note that selection criteria are case-sensitive. If you specify state='Alabama', IBM File Export will not select records where state='alabama'.

- 2) Typing **D** in the Cmd field next to a selection criterion deletes it from the list.
 - 3) Typing **A** on the Option line adds a line to the SQL Type panel on which you can specify additional selection criteria for the named type.
- c. If you select **S** on the Record Type List (ABX\$\$SQL) panel, the Select Statement List (AB X\$SELS) panel opens.

```

ABX$SELS  VIR1  -----  SELECT Statement List  -----  2004/12/20  12:11:19
Option  ==>                               Scroll ==> PAGE

Session Name      MDSTITIY
Record Type Name  DEFAULT

Cmd  Name                               Nil  Skip      Limit      Then

Valid Line Commands :  E - Edit SELECT Statement  D - Delete SELECT Statement
Valid Option Command:  A - Add New SELECT Statement

```

Figure 106. Select Statement List (ABX\$SELS) Panel

Panel columns:

Session Name

The name of the current session.

Record Type Name

The name of the record type, such as Debit or Credit.

Cmd Type any line command available on this panel in this column next to the desired segment.

Name

Lists the session name.

Nil

Specify Y to suppress writing the record that is produced by the SELECT statement. Specify N to indicate that writing of the record is not to be suppressed.

Skip

Type the number of records that you want to skip. You can specify that you want to skip the first 'nn' accepted records from the source when you are creating your output.

Limit

Type the number of records that is the maximum number you will produce in the target output.

Then

Type an integer in this field that specifies that after hitting the limit, 1 of the next 'n' acceptable records will be produced for the target.

Valid Line Commands:

E Type E to edit the selected SELECT statement.

D Type D to delete a SELECT statement.

Valid Option Commands:

- A Type A to add a new SELECT.
- 1) On this panel, you can
 - a) Type D on the Cmd line to delete a Select from the list.
 - b) Type E on a Cmd line to edit the Select
 - c) Type A on the Option line to add a new Select.
- 2) If you typed A or E on panel ABX\$SELS, the Edit Select Statement (ABX\$SELE) panel opens.

```
ABX$SELE  VIR1  -----  Edit Select Statement  -----  2004/12/20  12:24:38
Option  ==>>

Session Name

Record Type Name  DEFAULT

Statement Name

Generate Result Set Statement  N  (Y / N)

Data specification
  1  Field Specification
  2  Edit Selection Criteria
  3  Relationship Specification
```

Figure 107. Edit Select Statement (ABX\$SELE) Panel

Panel columns:

Session Name

The name of the current session.

Record Type Name

The name of the record type, such as Debit or Credit.

Statement Name

If this is a new Select statement, type its name here. If you are editing an existing statement, its name appears in this field.

Generate Result Set Statement

Specify Y to cause a Result Set statement to be generated as an output of Select processing, or N to suppress the generation of the Result Set Statement. See "RESULTSET" on page 236 for more information.

Valid Option Commands:

- 1 Type 1 to define or edit fields for this select statement.
- 2 Type 2 to create or edit selection criteria for this select statement.

- 3) Type **3** to specify information to define relationships that this Select statement will use.
 - 4) If you specified on the Session Settings panel that you wanted to issue SQL output, then this SQL Parameters Specification option is displayed. Type **4** to define SQL output parameters.
- 3) To define additional fields for a Select, entering **1** in the Cmd field results in opening panel ABX\$SLST.
 - 4) Panel ABX\$SLST displays those field that have been defined and permits you to add entries to or delete them from the list. The fields on ABX\$SLST are as follows:

```

ABX$SLST      ----- Selected Fields ----- 2004/08/05 15:21:33
Option  ==>                               Scroll ==> PAGE

      Session Name
      Record Type Name
Cmd  Field                               Type  Connected Field

Valid Line Commands : D - Delete Entry
Valid Option Commands: A - Add Entry

```

Figure 108. Selected Fields (ABX\$SLST) Panel

Panel fields:

Session Name

Displays the name of the session.

Record Type Name

Displays the record type name, such as Debit or Credit.

Cmd

Type any line command available on this panel in this column next to the desired segment.

Field

Displays the field name.

Type

Displays the type, for example, Debit or Credit.

Connected Field

When you include content data from an external source, you must specify the key field in order to connect to that content data. Type the name of that key field in this text entry field.

Valid Line Commands:

D

Type **D** to delete this entry from this panel.

Valid Option Commands:

A

Type **A** to add an entry to this panel.

- a) Typing **A** on the **Cmd** line next to an entry displays panel ABX\$FLST.

```

ABX$FLST      -----  New Field Selection  -----  2004/06/20  13:39:29
Option  ==>                               Scroll ==> PAGE

Session Name
Record Type Name

Cmd  Field                               Type

Valid Line Commands:  S - Select

```

Figure 109. New Field Selection (ABX\$FLST) Panel

Panel fields:

Session Name

Displays the name of the session.

Record Type Name

Displays the record type name, such as Debit or Credit.

Cmd Type any line command available on this panel in this column next to the desired segment.

Field Displays the field name.

Type On this panel, valid field Types may be **F** for Field, **V** for Variable, or **L** for Literal.

Valid Line Commands:

S Type **S** to select this entry from this panel and add it to ABX\$SLST. Press **Enter** to return to ABX\$SLST.

b) Type **S** in the Cmd field to select a field from this panel. Then press **Enter** to return to panel ABX\$SLST.

c) Press **PF3** to return to the Select Statements panel.

5) To define selection criteria, specifying **2** in the Cmd field of the Edit Select Statement (ABX\$SELE) panel results in opening panel ABX\$CLST.

```

ABX$CLST      -----  Selection Criteria  -----  2004/06/20  13:49:03
Option  ==>                               Scroll ==> PAGE

Session Name:

Row 1 of 2

Cmd  Value

Valid Line Commands:  D - Delete Line
Valid Option Commands:  A - Add Line

```

Figure 110. Selection Criteria (ABX\$CLST) Panel

Panel fields:

Session Name

Displays the name of the session.

Cmd Type any line command available on this panel in this column next to the desired segment.

Value

Type the value to be assigned as a selection criterion, for example, LITERAL=A=X'0100'. Note that selection criteria are case-sensitive. If you specify state='Alabama', IBM File Export will not select records where state='alabama'.

Valid Line Commands:

D Type **D** to delete an entry from the list. Press **PF3** to return to ABX\$SELS.

Valid Option Command:

A Type **A** to add a line on which you can specify additional selection criteria. Press **PF3** to return to ABX\$SELE.

- 6) To define relationships that this Select statement will use, specifying **3** in the Cmd field of the Edit Select Statement (ABX\$SELE) panel results in opening the Edit Relationship (ABX\$RRED) panel. This panel permits you to create or edit a relationship that will exist between the statement named in the **Statement Name** field and the statement you will name in the **Parent Statement Name** field.



Figure 111. Edit Relation (ABX\$RRED) Panel

Panel columns:

Session Name

The name of the current session.

Record Type Name

The name of the record type, such as Debit or Credit.

Statement Name

The name that you have assigned to this statement.

Parent Statement Name

Specify an identifier of up to 32 characters in length for the parent statement.

Valid Option Commands:

- 1 Type **1** to select a parent statement from a list of resultset statements on the Statement List (ABX\$RRLS) panel.
 - 2 Type **2** to create or edit relationship specifications that exist between the statements you have identified.
- 7) If you specified that you wanted to generate SQL output when defining your session settings, typing **Q** on a Cmd line of the Record Type List (ABX\$SRTL) panel or **4** on the Option line of the Edit Select Statement (ABX\$SELE) panel permits you to add SQL output parameters using the SQL Output Parameters panel, ABX\$SPFI. On ABX\$SPFI, you can provide a table name, table owner, and specify the number of records to be written out.
- 8) When you are finished making changes, continue to press **PF3** until you reach the IMS Menu.

Create DD statements

Follow these steps to create the DD statements required to generate the target output.

1. On the IMS Menu panel, select option **6, Create DD Statements**.
2. The DD Statements List (ABX\$IMSD) panel opens. On this panel, the name of your session is displayed in the **Session Name** field. A list of DD statements, created based on IMS parameters you have specified is displayed.

```
ABX$IMSD      ----- DD Statements List ----- 2004/10/21 11:27:45.44
Option  ==>>                               Scroll ==>> PAGE

Session Name

Row 1 of 2

Cmd  Label   Opt  Data Set Name
    IEFRDER
    ABXSQL

Valid Line Commands : E - Edit DD Statement
```

Figure 112. DD Statements List (ABX\$IMSD) panel

Panel fields:

Session Name

Displays the name of the session.

Cmd Type any line command available on this panel in this column next to the desired segment.

Label

The DD statement's name.

Opt Statements having a Y in this column are optional.

Data Set Name

The data set name associated with the DD statement.

Valid Line Command:

- E Type E on the Cmd line of any entry to edit the associated DD statements.
- 3. Complete the fields on this panel. To edit any DD statements, type E on the Cmd line to display the appropriate DD Statement Definition (ABX\$DDEn) panel, as, for example ABX\$DDE1 below.

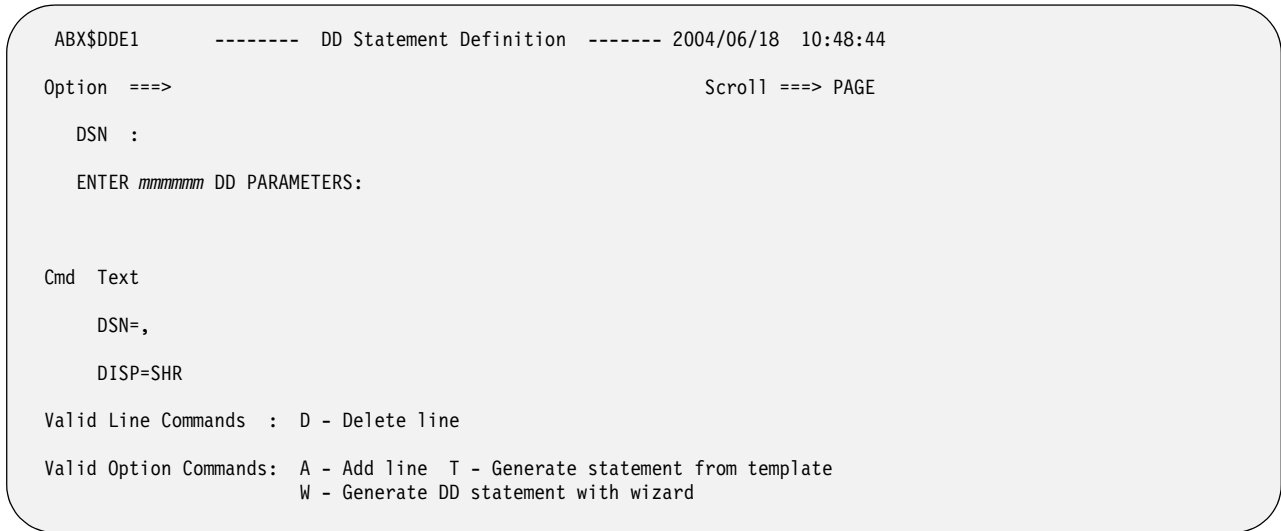


Figure 113. DD Statement Definition (ABX\$DDE1) panel

Panel fields:

DSN:

Type a data set name. Type a partially qualified name with an asterisk (*) on the line to open ABX\$DSL5, which displays a list of data set names from which you can choose the one you want to work with.

Valid Line Commands:

D Type D in the Cmd field delete a DD parameter.

Valid Option Commands:

A Type A on the Option line to add a line on which you can type another DD parameter.

T Type T on the Option line to use an existing data set definition as a template for your DD parameters.

W Type W on the Option line to use a wizard to create your DD parameters.

- 4. On this panel, you can add or delete DD statements, select a template to use to create DD statements, or use the Data Set Allocation Parameters Wizard (ABX\$DDWZ).

```

ABX$DDWZ      ---- Data Set Allocation Parameters ---- 2004/10/04 13:27
Option ==>>

Data Set unit          (If SMS managed, leave blank)

Volume serial          (If SMS managed, or unit specified
                        a tape unit, leave blank)

Space units            (CYL or TRK)

Primary quantity

Secondary quantity

Press ENTER to process, F12 to cancel

```

Figure 114. Data Set Allocation (ABX\$DDWZ) panel

Panel fields:

Data Set Unit:

Type the name of the data set device that will be used. Leave this field blank if this is an SMS-managed data set.

Volume Serial

Type the volume serial in this field. Leave this field blank if this is an SMS-managed data set or if you have specified a value for Data Set Unit.

Space Unit

Type **CYL** to use cylinders or **TRK** to use Tracks.

Primary Quantity

Type the number of tracks or cylinders to be allocated as the primary quantity.

Secondary Quantity

Type the number of tracks or cylinders to be allocated as the secondary quantity.

5. If you are writing out Fast Path data, the following DD statements are represented on the DD Statements List panel. The Fast Path SORTCTLS DD requires an additional panel to be completed as noted below.

FABCUR7: UR7DBDFN

The database DMB data set created by the UNLOAD process. If you edit this data set, you are taken to a DD Statement Definition panel on which you can add or delete DD statements.

FABCUR7: UR7DATA

The input data sets containing the segment data created by the UNLOAD process, FABCUR1. If you edit this data set, you are taken to a DD Statement Definition panel on which you can add or delete DD statements.

FABCUR7: UR7CTL

The control statement input stream for FABCUR7 (optional). If you edit this data set, you are taken to a DD Statement Definition panel on which you can add or delete DD statements, select a template to use to create DD statements, or use the Data Set Allocation Parameters Wizard (see Figure 78 on page 128).

FABCUR6: UR6DBDFN

The output data set containing the database DMB information that will be used by the DEDB reload program, FABCUR3. If you edit this data set, you are taken to a DD Statement Definition panel on which you can add or delete DD statements, select a template to use to create DD statements, or use the Data Set Allocation Parameters Wizard (see Figure 78 on page 128).

FABCUR6: UR6CTL

The control statement input stream for FABCUR6 (optional). If you edit this data set, you are taken to a DD Statement Definition panel on which you can add or delete DD statements.

FABCUR6: SORTCTLS

The XDnnnnnO and XSnnnnO pairs of data sets that contain the segment output and associated SORT control statements created by FABCUR6. If you edit this data set, you are taken to the Sort Control Statements panel (ABX\$XDXS).

```
ABX$XDXS      ----- Sort Control Statements ----- 2004/06/23 15:04:55
Option ==>>>                               Scroll ==>> PAGE

Session Name

Cmd  Number  XDStatemen  XSStatemen

Valid Line Commands : D - Delete statement
                    E - Edit XD statement  X - Edit XS statement
Valid Option Commands: A - Add Statement
```

Figure 115. Sort control statements (ABX\$XDXS) panel

Panel fields:

Session Name

Lists the name of the IBM File Export for z/OS session.

Cmd Type any line command available on this panel in this column next to the desired entry.

Number

A unique integer, generally assigned as a pair of statements is added, uniquely identifying a pair of XD/XS statements.

XD Statement

An XD statement defines an output data set for all of the database segment records produced for one or more of the areas defined in the DMB. For every XD statement you define, there must be a corresponding XS statement.

XS Statement

An XS statement defines an output data set that contains the Sort control statements for the segment data set associated with it. This data set is required even if the corresponding XD data set does not need to be sorted.

Valid Line Commands:

- D** Type **D** in the Cmd field to delete a pair of XD/XS statements.
- E** Type **E** in the Cmd field to edit an XD statement. Panel ABX\$DDE1 opens for this purpose (see Figure 77 on page 127).
- X** Type **X** in the Cmd field to edit an XS statement. Panel ABX\$DDE1 opens for this purpose (see Figure 77 on page 127).

Valid Option Commands:

- A** Type **A** in the Cmd field to add a line on which you can type another pair of XD/XS statements.

6. When you have completed your work, press **PF3** to return to the DD Statements List (ABX\$IMSD) panel. The information you provided is now filled in the appropriate fields. Pressing **PF3** again returns you to the IMS Menu.

Generate JCL

Follow these steps to generate the JCL required to generate the target output.

1. On the IMS Menu panel, select option 7, **Generate JCL**. Panel ABX\$IMSJ, JCL Generation for IMS Session, opens. On this panel, the name of your session is displayed in the **Session Name** field.
2. On this panel, if names are not already specified, specify or edit the name of the data set and member where the JCL will be saved.

```
ABX$IMSJ      --- JCL Generation for IMS Session ---
Option  ==>

Session Name

Enter data set and member name where jcl will be saved:
  Data set name
  Member name

Load Library Concatenations
  1 STEPLIBs
  2 RMODLIBs (optional)
  3 EXITLIBs (optional)
```

Figure 116. JCL Generation for IMS Session (ABX\$IMSJ) panel

Panel fields:

Session Name

Lists the name of the IBM File Export for z/OS session.

Data Set Name

If a name is not already supplied, type the fully-qualified name of the data set that will contain the member to which the JCL will be written in this field. You can browse for the data set by entering a partially qualified name with an asterisk.

Member

If a member name is not already specified, type the name of the member that will contain the JCL in this field. You may leave the field blank to see a member list. If you leave the field blank, panel ABX\$MEML opens to display the member list from which you can choose a member name.

Valid Option Commands:

- 1 Type 1 on the Option line to specify additional STEPLIBs.
- 2 Type 2 on the Option line to optionally specify additional RMODLIBs.
- 3 Type 3 on the Option line to optionally specify additional EXITLIBs.
3. Depending on the option you chose, panel ABX\$DSSL opens to display a list of existing data sets. You can add data sets to the list on this panel by typing A on the Option line. The panel displays a blank line on which you can type in the name of a data set to be added. When adding a data set, you can enter a partially qualified name and an asterisk. In this case, panel ABX\$DSLS is displayed, from which you can choose the data set you want to add.
4. To delete an existing data set from the panel, type D on the appropriate Cmd line to delete the specified data set from the list.

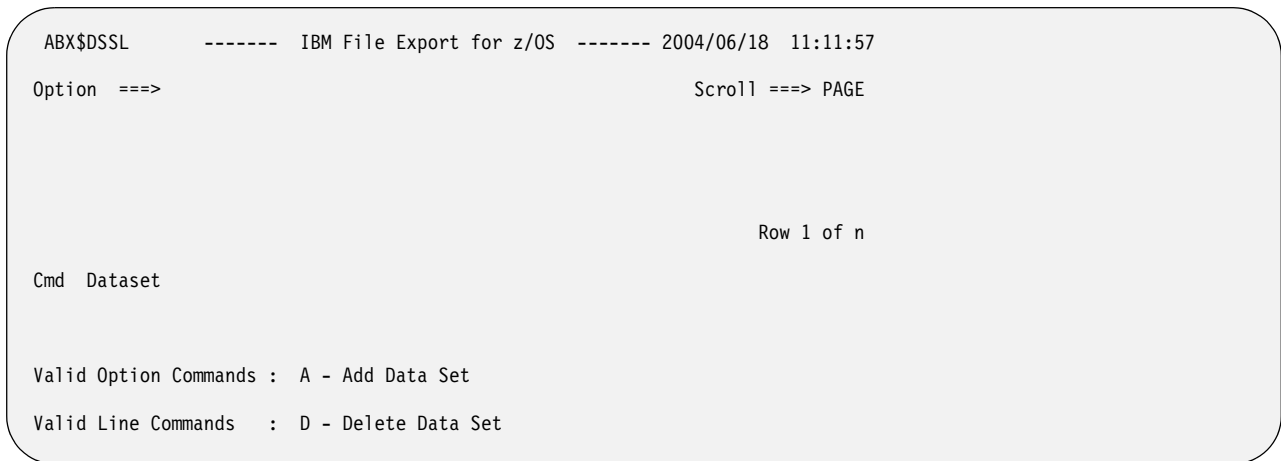


Figure 117. ABX\$DSSL panel

Panel fields:

Dataset:

Displays a list of data sets from which a data set can be deleted.

Valid Option Commands:

A Type A to add a line on which you may enter a data set name.

Valid Line Commands:

D Type D in the Cmd field to delete a data set name.

5. When you have completed work, press **PF3** to return to the JCL Generation for IMS Session (ABX\$FFJC) panel. The information you provided on the ABX\$DDEn panels is now filled in the appropriate fields.
6. Pressing **PF3** again returns you to the IMS Menu.

Chapter 9. Exporting Sequential Data or VSAM Data

This chapter explains how to use IBM File Export for z/OS to export sequential or VSAM data. The panels that are used in this process and their associated options, fields, and line commands are also explained. Note that the panels illustrated in this book are representative of the panels in the product and may differ slightly from the product panels.

Working with sequential data:

IBM File Export for z/OS can process sequential files as input and output. The supported record formats are: Fixed (F), Fixed Blocked (FB), Variable (V) and Variable Blocked (VB). The input file is designated by the ABXIN DD statement and may reside on DASD or TAPE. The output file is designated by the ABXOUT DD statement, and may also reside on DASD or TAPE.

The output record format is determined using a number of different methods. The highest precedence is the presence of a DCB= parameter coded in the JCL and attributed to the ABXOUT DD statement. This parameter may contain the RECFM and LRECL keywords to indicate what record format the output should be and the logical record length. The output data set may also be pre-allocated with the desired record format and logical record length. Lastly, if no record format or logical record length is specified, these attributes of the input file are used for the output file. In the case where no input file is supplied, as when the LOAD keyword is specified on the Global options panel.

You can use an input file containing fixed length records, and produce an output file containing variable length records. Conversely, you can also use an input file that contains variable length records and produce an output file containing fixed length records. The 4 byte RDW (Record Descriptor Word) will be added when building variable length records, and it will be removed, if necessary, when building fixed length records.

Note: The addition or deletion of the RDW may have adverse consequences on any data field alignment requirements.

Working with VSAM data:

IBM File Export for z/OS can process VSAM files as input and output. Key Sequenced Data Sets (KSDS), Entry Sequenced Data Sets (ESDS), Relative Record Data Sets (RRDS), and Variable Relative Record Data Sets (VRRDS) are all supported as input and output.

Summary

Select your sequential and VSAM tasks

“The Sequential and VSAM Menu” on page 174

Specify session settings and an SQL Output DD

“Specify session settings” on page 175

Define source and target data sets

“Specify source and target data sets” on page 178

Define variables and literals

“Define variables and literals” on page 181

Define relationships

“Define relationships” on page 183

Define field specification and manipulation

“Define field specification and manipulation” on page 187

Define type and select statement specification and manipulation

“Define type and select statement specification and manipulation” on page 191

Generate JCL

“Generate JCL” on page 198

The Sequential and VSAM Menu

Using IBM File Export, you can export data from sequential files to either sequential or VSAM files, and from VSAM files to either sequential or VSAM files.

From the IBM File Export Session Management panel, ABX\$SMNG (see page 102), select option 3, **z/OS Data Sets**, to provide specifications for a VSAM or sequential data set. You must specify a session name for the session you are creating. The Sequential and VSAM Menu, ABX\$SEQW, opens.

Note: If you want to work with an existing session, specifying 4 on the Option line of the IBM File Export Session Management (ABX\$SMNG) panel together with a session name in the Session Name field opens the specified session. Specifying 5 on the Option line displays a list of previously-defined sessions from which you can choose the one to work with.

```

ABX$SEQW      ----- Sequential & VSAM Menu ----- 2004/08/02 15:43:33
Option  ===>
Session Name: ABXSEQ2
Settings Specification
0  Session Settings
Data set Specification
1  Source & Target Data Set Specification
Data specification
2  Variable and Literal Specification
3  Relationship Specification
4  Field Specification
5  Type and Select statements Specification and Manipulation
JCL specification
6  Generate JCL

```

Figure 118. Sequential & VSAM Menu (ABX\$SEQW) panel

Panel fields:**Session Name**

Specifies the name of your current session.

Valid Option Commands:

- 0 Select this option to specify product options specific to this session. When you select this option, the Session Settings (ABX\$FOPT) panel opens.
- 1 Select this option to identify the source and target data sets you will use. When you select this option, the Source and Target Data Set Specification (ABX\$VSDB) panel opens.
- 2 Select this option to create variables and literals to be used during this session. When you select this option, the Variables and Literals (ABX\$VLST) panel opens.
- 3 Select this option to create external relationships to be used during this session. When you select this option, the Relationship Specification (ABX\$RELS) panel opens.
- 4 Select this option to define the specifications for field statements and how they should be manipulated. When you select this option, the Fields (ABX\$FSEQ) panel opens.
- 5 Select this option to define the specifications for type and select statements and how they should be manipulated. When you select this option, the Record Type List (ABX\$SQL) panel opens.
- 6 Select this option to display the JCL Generation (ABX\$JCLG) panel opens. On this panel, specify the information needed to generate the JCL to be used to generate the target output.

Specify session settings

1. From the Sequential and VSAM (ABX\$SEQW) panel, type **0** on the Option line. The Session Settings (ABX\$FOPT) panel opens.

```
ABX$FOPT V1R1 ----- Session Settings ----- 2004/11/04 16:46:57
Option ==>

Wild Card Char PERCENT      Decimal      DECIMAL
SQL Output      N (Yes / No)  SQL Qualifier

Skip
Limit
Then

Repository options
    Load all records automatically

Valid Option Commands: S - Specify SQL Output DD C - Connect to Repository
```

Figure 119. Session Settings (ABX\$FOPT) panel

Panel fields:

Wild Card Char

Type the character that you want to use as a wildcard in IBM File Export for z/OS in this field. The wild card is used in the LIKE predicate and represents the multi-character non-matching mask. Use this field to override global settings for this keyword.

Decimal

Type **Decimal** or **Comma** to specify the character to be used as the decimal point. Use this field to override global settings for this keyword.

SQL Output

Specify whether (Y) or not (N) you want to create SQL output. If you do not specify Y here, you will not see the SQL option on any of the subsequent panels for this session.

SQL Qualifier

Specify the qualifier to be used by SQL for unqualified objects.

Skip

Skip is defined as global in scope for Sequential and VSAM files. You can specify that you want to skip the first *nn* records from the source data when you are creating your output. Type the number of records to skip in this field.

Limit

Limit is defined as global in scope for Sequential and VSAM files. You can specify that you want to limit the number of eligible records that you output to the target to the value you specify here. Type the number of records to output in this field.

Then

Type a value specifying that, after the Limit value is reached, write to the output every 1 in *nn* records, where *nn* is the value you specify in this field. For example, if you specify a value of 10 for Then, IBM File Export will write out the number of records that meets the Limit value, then will write out every 10th record thereafter.

Load all records automatically

Place a slash on the line preceding this field if you want to have information from your default repository loaded automatically.

Valid Option Commands:

S

Type S on the Option line to specify information about the SQL Output DD

C

Type C on the Option line to open a connection to the repository.

2. Specify the settings to be in effect for this named session.
3. To specify information about the SQL Output DD, type S on the Option line. The DD Statement Definition panel opens.

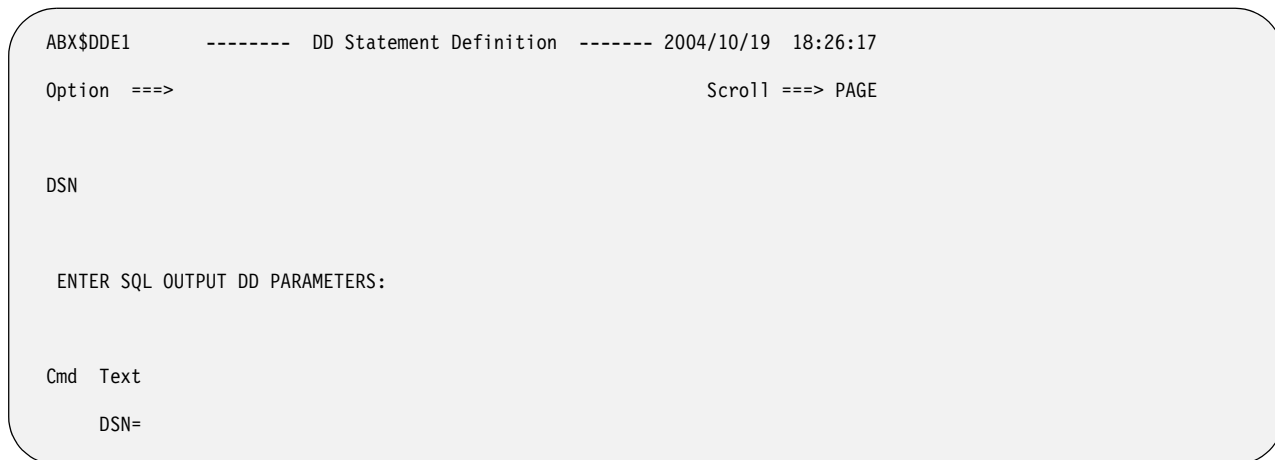


Figure 120. DD Statement Definition (ABX\$DDE1) panel

Panel fields:

DSN:

Type a data set name. Type a partially qualified name with an asterisk (*) on the line to open ABX\$DSL5, which displays a list of data set names from which you can choose the one you want to work with.

Valid Line Commands:

D Type **D** in the Cmd field delete a DD statement.

Valid Option Commands:

A Type **A** on the Option line to Add a DD parameter.

T Type **T** on the Option line to use an existing data set definition as a template for your DD parameters.

W Type **W** on the Option line to use the Data Set Allocation Parameters wizard.

4. If you typed **T** on the Option line, a pop-up window opens on which you can name the data set to be used as a template.
5. If you typed **W** on the Option line, the Data Set Allocation Parameters (ABX\$DDWZ) panel opens. On this panel, you can provide information used to create your DD statements.

```

ABX$DDWZ      ---- Data Set Allocation Parameters ---- 2004/10/04 13:27
Option  ==>>

Data Set unit          (If SMS managed, leave blank)

Volume serial          (If SMS managed, or unit specified
                        a tape unit, leave blank)

Space units            (CYL or TRK)

Primary quantity

Secondary quantity

Press ENTER to process, F12 to cancel

```

Figure 121. Data Set Allocation Parameters (ABX\$DDWZ) panel

Panel fields:

Data Set Unit:

Type the name of the data set device that will be used. Leave this field blank if this is an SMS-managed data set.

Volume Serial

Type the volume serial in this field. Leave this field blank if this is an SMS-managed data set or if you have specified a value for Data Set Unit.

Space Unit

Type **CYL** to use cylinders or **TRK** to use Tracks.

Primary Quantity

Type the number of tracks or cylinders to be allocated as the primary quantity.

Secondary Quantity

Type the number of tracks or cylinders to be allocated as the secondary quantity.

6. When you are through, pressing **PF3** multiple times moves you back through the panels and returns you to the Sequential and VSAM Menu panel.

Specify source and target data sets

1. From the Sequential and VSAM Menu (ABX\$SEQW) panel, type **1** on the Option line. The Sequential Data Set Specification panel (ABX\$VSDB) panel opens.


```

ABX$VSDB  V1R1 ----- Source and Target Data Set Specification -----
Option  ==>

Session Name  ABXSEQ2

Select Source Type and Enter Data Set Information:
Source Type   (VSAM / Sequential)
Source File
Source Member      (blank for sequential,
                   "*" for member selection list)

Select Target Type and Enter Data Set Information:
Target Type   (VSAM / Sequential)
Target File
Target Member      (blank for sequential,
                   "*" for member selection list)

Valid Option Commands:  S - Specify Source DD  T - Specify Target DD

```

Figure 122. Sequential Data Set Specification (ABX\$VSDB) panel

Panel fields:

Session Name

Lists the name of the IBM File Export for z/OS session.

Source Type

Type **V** for VSAM or **S** for sequential.

Source File

Type the name of the source file.

Source Member

If this is a sequential source file, type the name of the source member or type an asterisk in the field to browse for the member.

Target Type

Type **V** for VSAM or **S** for sequential.

Target File

Type the name of the target data set that you want to create.

Target Member

If this is a sequential target file, type the name of the target member or type an asterisk in the field to browse for the member.

Valid Option Commands:

- S** Use to display a panel on which you can add parameters to the DD statement designating the source file
- T** Use to display a panel on which you can add parameters to the DD statement designating the target file

2. Specify the type for the source file and target file. The type may be **V** for VSAM or **S** for sequential.
3. Type the name of the source data set in the **Source File** field.
4. Type the name of the target data set in the **Target File** field.
5. If either the source file or target file is a sequential file, specify a member name or enter an asterisk in the Source Member and/or Target Member field to see a member list.
6. To add parameters to the DD statements for the source files, or to add parameters to them, specify **S** on the Option line. The DD Statement Definition (ABX\$DDE4) panel opens.

```

ABX$DDE4      ----- DD Statement Definition ----- 2004/10/19  18:57:26
Option  ==>>                               Scroll ==>> PAGE

DSN =

ENTER SOURCE DD PARAMETERS:

Cmd  Text
     DSN=
Valid Line Commands :  D - Delete line
Valid Option Commands: A - Add line

```

Figure 123. DD Statement Definition (ABX\$DDE4) panel

Panel fields:

DSN:

This field displays the data set name you are working with.

Cmd Type any line command available on this panel in this column next to the desired entry.

Text Type a DD parameter in this field.

Valid Line Commands:

D Type **D** in the Cmd field delete a DD parameter.

Valid Option Commands:

A Type **A** in the Cmd field to add a line on which you can type another DD parameter.

7. To add parameters to the DD statements for the target files, or to add parameters to them, specify **T** on the Option line. Panel ABX\$DDE2 opens.

```

ABX$DDE2      ----- DD Statement Definition ----- 2004/06/21 11:59:18
Option  ==>>                               Scroll ==>> PAGE
      DSN  :

      ENTER TARGET DD PARAMETERS:

Cmd  Text
Valid Line Commands : A - Add line D - Delete line
Valid Option Commands: T - Generate statement from template
                        W - Generate DD statement with wizard

```

Figure 124. DD Statement Definition (ABX\$DDE2) panel

Panel fields:

- DSN** Lists the data set name you specified on the Sequential Data Sets Specification panel ABX\$VSDB.
- Cmd** Type any line command available on this panel in this column next to the desired entry.
- Text** The first line in this field cannot be changed and displays the text of the first DD statement. Subsequent lines may be added or deleted as you wish.

Valid Line Commands:

- A** Select to add a line to the DD statements you are creating.
- D** Select to delete a line from the DD statements you are creating.

Valid Option Commands:

- T** Select to use a template for the DD statements you are creating. If you enter T on the Option line, a pop-up panel opens on which you can enter the name of the data set you want to use as a template.

8. Pressing **PF3** on the DD Statement Definition panel returns you to the Sequential Data Sets Specification (ABX\$VSDB) panel Pressing **PF3** again returns you to the Sequential and VSAM Menu.

Define variables and literals

Follow these steps to define or edit variables and literals. Literals are constants and can be either implicit or named. Variables are literals that change value during the process due to being incremented as a result of system action.

1. On the Sequential and VSAM Menu panel, select option **2, Variable and Literal Specification**. Panel ABX\$VLST, containing a list of any already-defined variables and literals, opens. On this panel, the name of your session is displayed in the **Session Name** field. Any variables or literals you have previously defined are displayed.
2. To define a variable, type **AV** on the Option line. To define a literal, type **AL** on the Option line.

```

ABX$VLST      ----- Variables and Literals ----- 2004/06/18 09:25:00
Option ==>>                               Scroll ==> PAGE

Session Name

Row 1 of 2

Cmd  Label                               Type
TESTVAR                               L
TESTLIT                               V

```

Figure 125. Variables and Literals (ABX\$VLST) panel

Panel fields:

Session Name

Lists the name of the IBM File Export for z/OS session.

Cmd Type any line command available on this panel in this column next to the desired entry.

Label

Displays the name you have assigned to this variable or literal.

Type Displays **L** for literal or **V** for Variable.

Valid Option Commands:

AL Type **AL** on the Option line to define a literal.

AV Type **AV** on the Option line to define a variable.

Valid Line Commands:

E Type **E** on a line to edit the associated literal or variable.

D Type **D** on a line to delete the associated literal or variable.

3. If you have entered **AL** or **AV** on the Option line, the Variables and Literals panel (ABX\$VARS) displays. On this panel, you can define a new variable or literal or edit an existing variable or literal.

```

ABX$VARS      ----- New Variable ----- 2004/06/18 09:41:42
Option ==>>

Source Name

Length      0
Data Type   CHAR

Value

Press ENTER or F3 to process, F12 to cancel

```

Figure 126. New Variable (or Literal) (ABX\$VARS) panel

Panel fields:

Source Name

Type a name for the source data.

Length

Type the length of the source data in this field. Length is a numeric literal and is the length in bytes required in memory for the source data. The sum of offset and length cannot exceed the record length.

Data Type

Type the data type in this field. (For a description of these data types, see Table 3 on page 257.) The data type can be one of the following:

- INT or INTEGER
- CHAR
- FLOAT
- HEX
- SHORT
- DOUBLE
- PACKED

Value

Type the value to be assigned to the variable or literal, for example, X'0100'.

4. When you have completed work on this panel, press **PF3** to return to the Variables and Literals List. Pressing **PF3** again returns you to the Sequential and VSAM Menu.

Define relationships

Follow these steps to provide information to define external relationships. Before doing so, you may want to review the information about application-defined relationships in “Defining relationships in IBM File Export” on page 5.

1. On the Sequential and VSAM Menu panel, select option **3, Relationship Specification**. Panel ABX\$RELS opens.

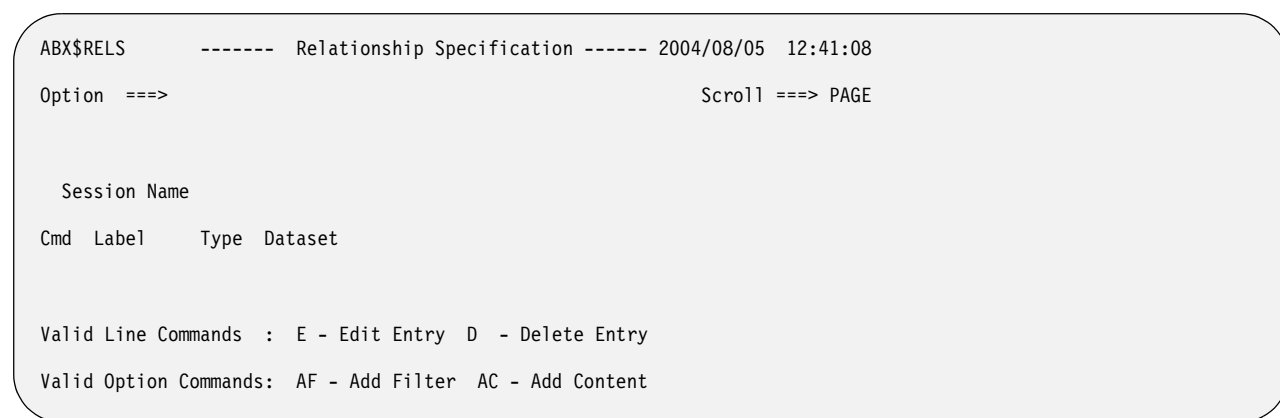


Figure 127. Relationship Specification (ABX\$RELS) panel

Panel fields:

Session Name

Lists the name of the IBM File Export for z/OS session under which you are working.

Panel columns:

Cmd Type any line command available on this panel in this column next to the desired entry.

Label

Lists the name you have assigned to this filter or content specification.

Type Lists an **F** if this is a filter or **C** if this is a content specification.

Data Set

Names the data set with which filter or content relationship data has been associated.

Valid Line Commands:

E Use to edit an entry in this list. When you type the **E** line command in the **Cmd** field, the ABX\$RELF (if for a filter specification) or ABX\$RELC (if for a content specification) panel opens.

D Use to delete an entry from this list.

Valid Option Commands:

AF Use to add a filter to the current data set. When you type the **AF** option, the ABX\$RELF panel opens.

AC Use to add a content specification to the current data set. When you type the **AC** option, the ABX\$RELC panel opens.

2. If you entered AF to add a filter, the Filter Specification panel (ABX\$RELF) opens. Use this panel to add a filter. The name of the source data set that contains the related data that you want to add to your target is displayed in the Source Name field if you have selected that data set from the External Data Relationships list. Default values appear in the rest of the fields. You can change the value in any field by overtyping it.

Note: External data is that data contained in a data set from which source data will be selected to be exported to the target, as, for example, a VSAM file you want to export into your target table, or a sequential file from which you want to pull fields to be added to your target object.

```
ABX$RELF ----- Filter Specification ----- 2004/06/20 14:39:58
```

Source Name

DSN

Position 0

Length 0

Data Type HEX

Press ENTER or F3 to process, F12 to cancel

Figure 128. Filter Specification (ABX\$RELF) panel

Panel fields:

Source Name

Type the name you are assigning to this filter.

DSN

Type the name of the data set that is the source for this filter. If you put an asterisk in this field, panel ABX\$DSLS opens. On this panel, you can select the data set you want to use.

Position

Type the position of the source data in this field. Position can be specified:

- As a numeric integer type constant like 1 or 100
- Relative to the end of the record like 'end - 10' (ten bytes from the end of the record).
- Using a hex constant (x'22' or h'22') which will always be interpreted as an offset (never as a 'position').

Length

Type the length of the source data in this field. Length is a numeric literal. It is the length of the bytes required in memory for the source data.

Data Type

Type the data type of the source data set in this field. (For a description of these data types, see Table 3 on page 257.) You can specify any of the following types:

- INT
- CHAR
- FLOAT
- HEX
- SHORT
- DOUBLE
- PACKED

Pressing **Enter** returns you to the Relationship Specification panel, ABX\$RELS.

3. If you entered AC to add a content specification, the Relation Content panel (ABX\$RELC) opens.

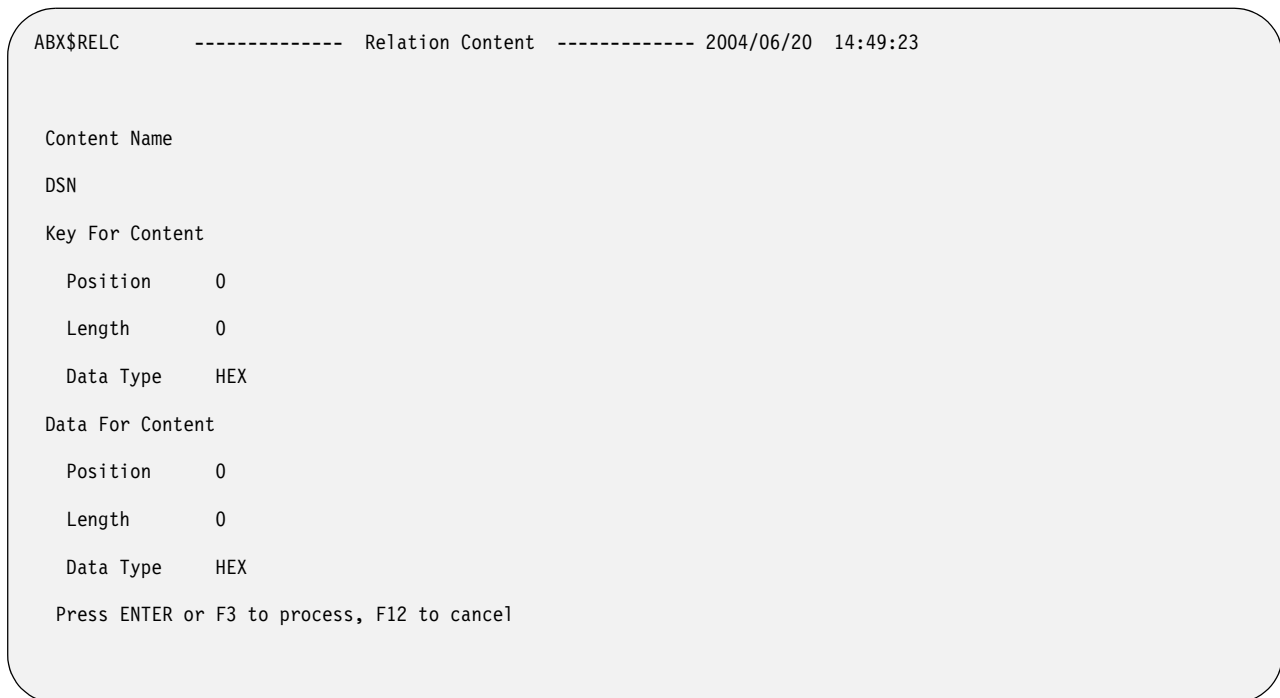


Figure 129. Relation Content (ABX\$RELC) panel

Panel fields:

Content Name

Type the name you are assigning to this content specification.

DSN Type the name of the data set that is the source for this content specification. If you type a partially qualified name with an asterisk in this field, panel ABX\$DSLS opens. On this panel, you can select the data set you want to use.

Position

Type the position of the source data in this field. Position can be specified:

- As a numeric integer type constant like 1 or 100
- Relative to the end of the record like 'end - 10' (ten bytes from the end of the record).
- Using a hex constant (x'22' or h'22') which will always be interpreted as an offset (never as a 'position').

Length

Type the length of the content key or content data (as appropriate) in this field. Length is a numeric literal and is the length of the bytes required in memory for the source data.

Data Type

Type the data type of the source data in this field. (For a description of these data types, see Table 3 on page 257.) You can specify any of the following types:

- INT
- CHAR
- FLOAT
- HEX
- SHORT
- DOUBLE

- PACKED

Note: The position, length, and data type fields are repeated first for the key and second for the content specification.

Use this panel to specify information about the content data set. The DSN field is filled in for you if you have selected the data set from the External Data Relationships list. Default values appear in the rest of the fields. You can change the value in any field by overtyping it.

To add content, you select data from a data set and match that data to the data from a specific column in the source in order to populate your target. Use the key to identify the key column. Preferably, that column is unique, as, for example, Account Number.

Use the Data to identify actual data that is to be selected from the source to be written to the target.

To summarize, use a content specification to include data from an external source. Use the Key to specify what data IBM File Export is to compare to the source file, and the Data to specify what data IBM File Export is to include in the target output when the key is found.

For example, assume that your source file contains addresses using two-character State abbreviations. But you would rather have addresses with the State name spelled out in the target data. To accomplish this, you would define a Content Relationship in File Export to use an external file that contains the abbreviated State name in columns 1 and 2, and the spelled out State name in columns 3 through 28. In this example, the Key Column in the external file would be the abbreviated State name (since the same data exists in the source file) in columns 1 and 2, and the Data Column that you want to include in the target data would be the data contained in columns 3 through 28. So that when File Export finds 'TX' in the source file as the abbreviated State name, for example, the Key from the external file is matched, and File Export uses the spelled-out State name from the external file to populate the target data file.

Note: External data is that data contained in a data set from which source data will be selected to be exported to the target, as, for example, a VSAM file you want to export into your target table, or a sequential file from which you want to pull fields to be added to your target object.

Pressing **PF3** returns you to the Relationship Specification panel, (ABX\$RELS).

4. Entering **E** on the Cmd line next to a filter or content specification, opens the ABX\$RELF (if for a Filter specification) or ABX\$RELC (if for a content specification) panel. When you have completed your work on this panel, pressing **Enter** returns you to the Relationship Specification panel (ABX\$RELS).
5. Press **PF3** to return to the Sequential and VSAM Menu (ABX\$SEQW).

Define field specification and manipulation

1. Selecting **4** on the Sequential and VSAM Menu permits you to define field specification and manipulation for this session. IBM File Export for z/OS opens the Fields panel, ABX\$FSEQ.

```

ABX$FSEQ ----- Fields ----- 2004/06/22 10:13:54
Option ==>                               Scroll ==> PAGE

Session Name:

File:

Max Size:      0

Cmd  Field                Start  Size  Type  Scramble

Valid Line Commands : E - Edit Field D - Delete Field
Valid Option Commands: A - Add Field

```

Figure 130. Fields (ABX\$FSEQ) panel

Panel fields:

Session Name

Lists the name of the IBM File Export for z/OS session under which you are working.

Cmd Type any valid line command in the Cmd field.

File Lists the name of the source data set.

Max Size

This field is pre-filled with the maximum logical record length as determined by IBM File Export for z/OS.

Column entries:

Field This field is pre-filled with the current field name.

Start This field is pre-filled with the current start position.

Size This field is pre-filled with the current field size.

Type This field is pre-filled with the current data type. (For a description of these data types, see Table 3 on page 257.) You can specify any of the following data types:

- INT or INTEGER
- CHAR
- FLOAT
- HEX
- SHORT
- DOUBLE
- PACKED

Scramble

Type **Y** in this field to indicate that this field should be scrambled. Leave the field blank or type **N** to indicate that the field should not be scrambled. When a column is scrambled, the contents will retain the same format as in the source, but the contents will be randomized. This is useful when the source contains sensitive data. The algorithm used for scrambling produces reproducible output. This means that the output of a scrambled value will be the same each time that value is scrambled.

Valid Line Commands:

- E** Select to edit a field on the panel.
- D** Deletes the field from the currently defined list of fields.

Valid Option Commands:

- A** Select to add a field to the panel.
2. If you have entered **A** to add a field, or **E** to edit a field, the Field Specification panel, ABX\$NFLD, opens.



Figure 131. Field Specification (ABX\$NFLD) Panel

Panel fields:**Field Name**

Type a name for the field to be added to the segment.

Field Position

Type the position where the source data starts. Position can be specified:

- As a numeric integer type constant like 1 or 100
- Relative to the end of the record like 'end - 10' (ten bytes from the end if the record).
- Using a hex constant (x'22' or h'22') which will always be interpreted as an offset (never as a 'position').
- As a cumulative position/offset definition. For example:
 - Use the NEXT declarative to specify that the position is the next character after the prior field.
 - Use the NEXTC declarative to specify that the position is the next character after the prior field.
 - Use the NEXTI declarative to specify that the position is the next integer position after the prior field.
 - Use the NEXTW declarative to specify that the position is the next word position after the prior field.
 - Use the NEXTH declarative to specify that the position is the next halfword location after the prior field.
 - Use the NEXTD declarative to specify that the position is the next double word location after the prior field.

Information: NEXTx does NOT set the field attribute (WORD/DOUBLE) just the position.

Field Length

Type the field length of the field.

Field Type

Type the data type for the field. (For a description of these data types, see Table 3 on page 257.) The data type may be one of the following:

- INT or INTEGER
- CHAR
- FLOAT
- HEX
- SHORT
- DOUBLE
- PACKED

Occurs

Type an integer that indicates the number of times a variable length repeating group occurs.

Depending On

Type the object of the Occurs. If the number of times a repeating group can occur in the record depends on this object of the Occurs, that dependency should be specified here. For information about Occurs Depending On, see "Understanding Occurs Depending On" on page 262.

Precision

Type the precision of the numeric data.

Scale Type the scale of the numeric field.

Signed

Specify whether or not the numeric data is signed.

Scramble

Type **Y** in this field to indicate that this field should be scrambled. Leave the field blank or type **N** to indicate that the field should not be scrambled. When a column is scrambled, the contents will retain the same format as in the source, but the contents will be randomized. This is useful when the source contains sensitive data. The algorithm used for scrambling produces reproducible output. This means that the output of a scrambled value will be the same each time that value is scrambled.

Group Length

Type the length of the group.

Maximum Length

Type the maximum length of the field.

Position Algorithm

Because the length of a group item containing variable length data is not a constant, an algorithm is required to express the position of a field, and in some cases, the length of the field or group item. Type the position algorithm in this field.

Group Length Algorithm

Because the length of a group item containing variable length data is not a constant, an algorithm is required to express the position of a field, and in some cases, the length of the field or group item. Type the group length algorithm in this field.

Enter "/" to select option

To specify an additional position algorithm, group length algorithm, or both, type a slash (/) in the appropriate field or fields.

3. Make any necessary modifications to the fields on the Field panel by overtyping the default values. Then press **PF3** to exit ABX\$NFLD and return to the Fields panel, ABX\$FSEQ. Pressing **PF3** again returns you to the Sequential and VSAM Menu.

Define type and select statement specification and manipulation

1. Selecting **5** on the Sequential and VSAM menu permits you to define type and select statement specification and manipulation for this session. IBM File Export for z/OS opens the Record Type List panel (ABX\$SQTL).
2. Use panel ABX\$SQTL to add a new type, edit the types for the file, delete an existing type, or to specify fields and selection criteria for a type. If you specified on the Session Settings panel that you wanted to generate SQL output, you can also define SQL output parameters using a command on this panel.

```

ABX$SQLT ----- Record Type List ----- 2004/08/04 10:31:27
Option ==>>                               Scroll ==> PAGE
Session Name ABXIMS01

Row 1 of 2

Cmd Name                               Excl Skip      Limit      Then
SIZE                                    N
DEFAULT                                 N

Valid Line Commands : E - Edit Type  D - Delete Type  F - Field List
                    S - SELECT statement list

Valid Option Command: A - Add New SQL Type  L - Load from Repository

```

Figure 132. Record Type List (ABX\$SQLT) Panel

Panel fields:

Session Name

Lists the name of the IBM File Export for z/OS session under which you are working.

Cmd Type any valid line command in the Cmd field.

Name

Lists the name you have assigned to the field.

Excl If this field contains a **Y**, it is excluded when selecting fields for processing. If it contains an **N**, the field is included when selecting fields for processing.

Skip Type the number of records that you want to skip. You can specify that you want to skip the first 'nn' accepted records from the source when you are creating your output.

Limit

Type the number of records that is the maximum number you will produce in the target output.

Then Type an integer in this field that specifies that after hitting the limit, 1 of the next 'n' acceptable records will be produced for the target.

Valid Line Commands:

E Type **E** in the Cmd field to edit the selected type.

D Type **D** in the Cmd field to delete the selected type.

F Type **F** to display a list of fields.

S Type **S** in the Cmd field to create select statements for a type.

Q If you specified on the Session Settings panel that you want to issue SQL output, you will have the **Q** line command available to you on this panel. Use the **Q** command to specify SQL output parameters.

Valid Option Commands:

A Type **A** on the Option line to add a new SQL type.

L Type **L** to load data from the repository.

a. If you selected **A** to add a new type to this panel, or **E** to edit a given type, the Record Type panel (ABX\$SQLT) opens. On this panel, you can provide a name and value for a new field or edit the name and value for an existing field.

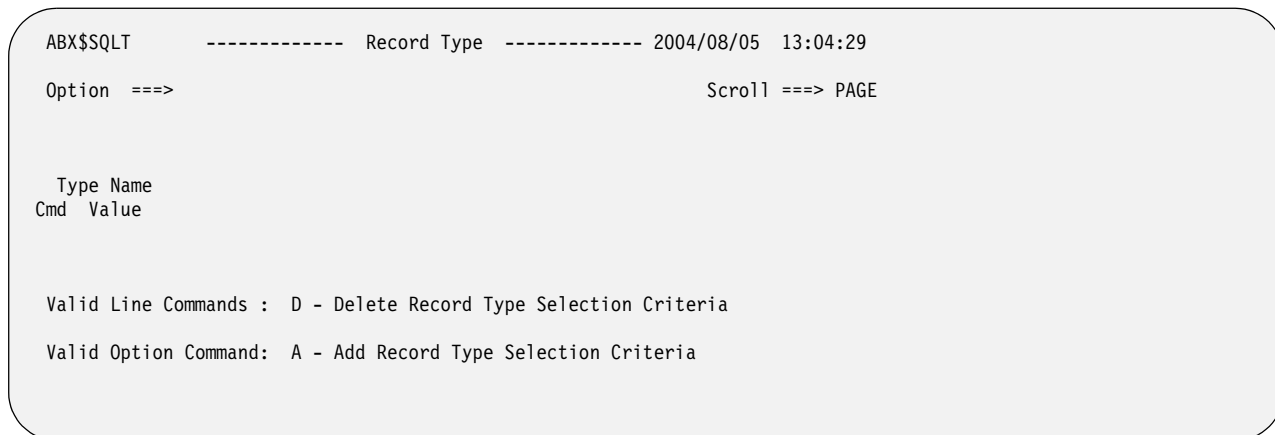


Figure 133. Record Type (ABX\$SQLT) panel

Panel fields:

Type Name

Type the name of the field, for example, Debit or Credit, for which you want to specify a condition.

Cmd Type any line command available on this panel next to the desired entry.

Value

Type a value in the field, which might be in the form of a WHERE predicate.

Valid Line Commands:

D Type **D** in the Cmd field to delete the selection criterion.

Valid Option Commands:

A Type **A** on the Option line to add a new selection criterion.

- 1) On the Record Type (ABX\$SQLT) panel, you can add or delete selection criteria for the type you have added or are editing. Note that selection criteria are case-sensitive. If you specify state='Alabama', IBM File Export will not select records where state='alabama'.
 - 2) Typing **D** in the Cmd field next to a selection criterion deletes it from the list.
 - 3) Typing **A** on the Option line adds a line to the SQL Type panel on which you can specify additional selection criteria for the named type.
- b. If you select **S** on the Record Type List (ABX\$SQLT) panel, the Select Statement List (AB X\$SELS) panel opens.

```

ABX$SELS  VIR1 ----- SELECT Statement List ----- 2004/12/20 12:11:19
Option  ==>                               Scroll ==> PAGE

Session Name      MDSTITIY
Record Type Name  DEFAULT

Cmd  Name                Nil  Skip      Limit      Then

Valid Line Commands :  E - Edit SELECT Statement  D - Delete SELECT Statement
Valid Option Command:  A - Add New SELECT Statement

```

Figure 134. Select Statement List (ABX\$SELS) Panel

Panel columns:

Session Name

The name of the current session.

Record Type Name

The name of the record type, such as Debit or Credit.

Cmd Type any line command available on this panel in this column next to the desired segment.

Name

Lists the session name.

Nil

Specify Y to suppress writing the record that is produced by the SELECT statement. Specify N to indicate that writing of the record is not to be suppressed.

Skip

Type the number of records that you want to skip. You can specify that you want to skip the first 'nn' accepted records from the source when you are creating your output.

Limit

Type the number of records that is the maximum number you will produce in the target output.

Then

Type an integer in this field that specifies that after hitting the limit, 1 of the next 'n' acceptable records will be produced for the target.

Valid Line Commands:

E Type E to edit the selected SELECT statement.

D Type D to delete a SELECT statement.

Valid Option Commands:

A Type A to add a new SELECT.

1) On this panel, you can

a) Type **D** on the Cmd line to delete a Select from the list.

b) Type **E** on a Cmd line to edit the Select

- c) Type **A** on the Option line to add a new Select.
- 2) If you typed **A** or **E** on panel ABX\$SELS, the Edit Select Statement (ABX\$SELE) panel opens.

```

ABX$SELE  V1R1  -----  Edit Select Statement  -----  2004/12/20  12:24:38

Option  ==>

Session Name

Record Type Name  DEFAULT

Statement Name

Generate Result Set Statement  N  (Y / N)

Data specification

1  Field Specification

2  Edit Selection Criteria

3  Relationship Specification

```

Figure 135. Edit Select Statement (ABX\$SELE) Panel

Panel columns:

Session Name

The name of the current session.

Record Type Name

The name of the record type, such as Debit or Credit.

Statement Name

If this is a new Select statement, type its name here. If you are editing an existing statement, its name appears in this field.

Generate Result Set Statement

Specify **Y** to cause a Result Set statement to be generated as an output of Select processing, or **N** to suppress the generation of the Result Set Statement. See "RESULTSET" on page 236 for more information.

Valid Option Commands:

- 1 Type **1** to define or edit fields for this select statement.
 - 2 Type **2** to create or edit selection criteria for this select statement.
 - 3 Option **3** is not available for Sequential or VSAM processing.
 - 4 If you specified on the Session Settings panel that you wanted to issue SQL output, then this SQL Parameters Specification option is displayed. Type **4** to define SQL output parameters.
- 3) To define additional fields for a Select, entering **1** in the Cmd field results in opening panel ABX\$SLST.

- 4) Panel ABX\$SLST displays those field that have been defined and permits you to add entries to or delete them from the list. The fields on ABX\$SLST are as follows:

```
ABX$SLST      ----- Selected Fields ----- 2004/08/05 15:21:33
Option  ==>                               Scroll ==> PAGE

Session Name
Record Type Name

Cmd  Field                               Type  Connected Field

Valid Line Commands : D - Delete Entry
Valid Option Commands: A - Add Entry
```

Figure 136. Selected Fields (ABX\$SLST) Panel

Panel fields:

Session Name

Displays the name of the session.

Record Type Name

Displays the record type name, such as Debit or Credit.

Cmd Type any line command available on this panel in this column next to the desired segment.

Field Displays the field name.

Type Displays the type, for example, Debit or Credit.

Connected Field

When you include content data from an external source, you must specify the key field in order to connect to that content data. Type the name of that key field in this text entry field.

Valid Line Commands:

D Type **D** to delete this entry from this panel.

Valid Option Commands:

A Type **A** to add an entry to this panel.

a) Typing **A** on the Cmd line next to an entry displays panel ABX\$FLST.

```

ABX$FLST      ----- New Field Selection ----- 2004/06/20 13:39:29
Option  ==>                               Scroll ==> PAGE

Session Name
Record Type Name

Cmd  Field                Type

Valid Line Commands:  S - Select

```

Figure 137. New Field Selection (ABX\$FLST) Panel

Panel fields:

Session Name

Displays the name of the session.

Cmd Type any line command available on this panel in this column next to the desired segment.

Field Displays the field name.

Type On this panel, valid field Types may be F for Field, V for Variable, or L for Literal.

Command Line Entries:

S Type **S** to select this entry from this panel and add it to ABX\$SLST. Press **PF3** to return to ABX\$SLST.

- b) Type **S** in the Cmd field to select a field from this panel. Then press **Enter** to return to panel ABX\$SLST.
 - c) Press **Enter** to return to the Select Statements panel.
- 5) To define selection criteria, specifying **2** in the Cmd field of the Edit Select Statement (ABX\$SELE) results in opening panel ABX\$CLST.

```

ABX$CLST      ----- Selection Criteria ----- 2004/06/20 13:49:03
Option  ==>                               Scroll ==> PAGE

Session Name:

Row 1 of 2

Cmd  Value

Valid Line Commands:  D - Delete Line
Valid Option Commands: A - Add Line

```

Figure 138. Selection Criteria (ABX\$CLST) Panel

Panel fields:

Session Name

Displays the name of the session.

Cmd Type any line command available on this panel in this column next to the desired segment.

Value

Type the value to be assigned as a selection criterion, for example, LITERAL=A=X'0100' . Note that selection criteria are case-sensitive. If you specify state='Alabama', IBM File Export will not select records where state='alabama'.

Valid Line Commands:

D Type **D** to delete an entry from the list. Press **PF3** to return to ABX\$SELS.

Valid Line Commands:

A Type **A** to add a line on which you can specify additional selection criteria. Press **PF3** to return to ABX\$SELS.

- c. If you specified that you wanted to generate SQL output when defining your session settings, typing **Q** on a Cmd line of the Record Type List panel, ABX\$SQL, or **4** on the Option line of the Edit Select Statement (ABX\$SELE) panel permits you to add SQL output parameters using the SQL Output Parameters panel, ABX\$SPFI. On ABX\$SPFI, you can provide a table name, table owner, and specify the number of records to be written out.
- d. When you are finished making changes, continue to press **PF3** until you reach the Sequential & VSAM Menu (ABX\$SEQW) panel.

Generate JCL

1. To generate JCL containing your exported data, on the Sequential & VSAM Menu (ABX\$SEQW) panel, select **6, Generate JCL**. Panel ABX\$JCLG, the JCL Generation panel, opens.

```

ABX$JCLG      ----- JCL Generation -----
Option  ==>

Session Name:

Enter data set and member name where jcl will be saved

Data set name:

Member name:      (Blank for member list)
Enter AL command to add more STEPLIBs
  
```

Figure 139. JCL Generation (ABX\$JCLG) panel

Panel fields:**Session Name**

Lists the name of the IBM File Export for z/OS session.

Data set name:

If the data set name is not displayed when this panel opens, type the name of the data set in which the JCL will be saved.

Member name:

If the member name is not displayed when this panel opens, type the name of the member in which the JCL will be saved. If you don't know the member name, leave the field blank and press **Enter** to view a list of members.

Valid Option Commands:

AL Type **AL** on the Option line to add more STEPLIBs to your JCL definition.

2. On this panel, if the Data Set Name and Member Name are not displayed when the panel opens, type these names to indicate where you want to store the generated JCL. If you want to browse for the data set name, enter a partially qualified name and press **Enter**. IBM File Export for z/OS opens panel ABX\$DLSL which displays a list of data sets from which you can choose.
3. If you left the member name blank, panel ABX\$MEML displays a list of members from which you can choose the member to use by entering **S** in the Cmd field. After you have selected a member from this display, pressing **Enter** returns you to the JCL Generation panel.



Figure 140. Member Selection List (ABX\$MEML) panel

Panel fields:

Cmd Type **S** in this column next to the desired entry.

Member Name

Displays a list of members from which you can choose the member whose name will be filled in on the JCL Generation panel.

Valid Option Commands:

S Type **S** in the Cmd field to select this entry.

4. To add more STEPLIBs to your generated JCL, type **AL** on the Option line to display panel ABX\$DSSL.

```
ABX$DSSL      ----- IBM File Export for z/OS ----- 2004/06/21 14:25:36
Option  ==>                               Scroll ==> PAGE

Cmd  Dataset

Valid Option Commands : A - Add Data Set
Valid Line Commands   : D - Delete Data Set
```

Figure 141. ABX\$DSSL panel

Panel fields:

Cmd Type any line command available on this panel in this column next to the desired entry.

Data set

Displays a list of data sets from which you can choose additional STEPLIBs.

Valid Option Commands:

A Type **A** on the Option line to add a data set to this panel..

Valid Line Commands:

D Type **D** on the Cmd line to delete a data set from the list.

5. Once you have completed work on panel ABX\$DSSL, press **PF3** to return to the JCL Generation (ABX\$JCLG) panel. Press **Enter** to generate your JCL.
6. Pressing **PF3** again returns you to the Sequential and VSAM Menu (ABX\$SEQW).

Part 6. Using Control Statements

Chapter 10. Creating JCL Using Control

Statements	203
Control statement syntax	203
General order of control statements	203
IMS example	203
Names used in control statements	204
Applicability of control statements based on type of source data	204
Control statement syntax, definitions, and descriptions	206
AGN	207
ALTID	207
ALTRRC00	207
AUDIT	207
BKO	208
BUF	208
COLLECT	208
COLUMN	209
CPUTIME	210
DATE	210
DBDLIB	212
DBRC	212
DFSVSAMP	213
DIRCA	213
DLTDUPS	213
DOPTLIB	214
DPSBMAC	214
DPSBPFX	214
DPSBSFX	215
DYNPSB	215
EXCLUDE	215
EXCPVR	216
EXITACTIVE	216
EXITNAME	216
FIELD	217
Examples	219
FMTO	219
GSGNAME	220
IMSID	220
IMSPLEX	220
INPCB#	221
INSERT	221
IRLM	221
IRLMNAME	222
LAST_SEGMENT	222
LIMIT	222
LITERAL	222
LOCKMAX	224
MON	224
NBA	224
OBA	225
OFIELD	225
Examples	227

OPT	228
OPTIONS	228
OUTPCB#	232
PARDLI	232
PROCOPT	232
PSBNAME	233
RELATIONSHIP	233
RESLIB1	236
RESLIB2	236
RESLIB3	236
RESULTSET	236
Parent-child relationships	237
SEGMENT	237
SELECT statements	239
WHERE	240
Using the Asterisk (*)	244
SKIP	244
SOURCEDB2	244
SOURCEDBD	244
SOURCEDD	245
SSM	245
STIMER	245
SWAP	246
TABLE	246
TARGETDB2	248
TARGETDBD	248
THEN INSERT	248
Requirements associated with the THEN INSERT statement in IBM File Export for z/OS	251
TYPE	251
VARIABLE	254
WUNIT	256

Chapter 11. Additional Information about Control Statement Syntax

Including comments in your JCL	257
About scrambling	257
Specification of IBM File Export data types	257
Understanding scope	257
Scope of fields, literals, and variables	258
Scope of SKIP and LIMIT	258
Limiting output record counts using SKIP and LIMIT	258
Global specification of SKIP and LIMIT	258
Non-global specification of SKIP and LIMIT	259
THEN ONE IN mmm	259
Inserting data into IMS databases using the File Export UNLOAD file	259
About ABXFELD0	259
Requirements and restrictions	259
JCL requirements	260
Example JCL	260
ABXFELD0 control statements	260

OPTIONS control statement	260	Understanding Occurs Depending On	262
EXCLDBD control statement	260	Specifying MINIMUM, MAXIMUM, PRINT, or	
INCLDBD control statement	261	SPUFI	262
Working with variable length records	261	Chapter 12. Formatting Output Records	263
Variable length input records and the use of the		Output record formatting	263
asterisk (*)	261	How use of the Cobol parser affects output	
Using slack bytes when creating output files . . .	261	alignment	264
Understanding INSERT and THEN INSERT . . .	262		

Control statements are used to specify information about the source data to be extracted and target data to be created by IBM File Export for z/OS. Using these control statements permits you to create JCL directly rather than through the use of the ISPF user interface. This part describes the syntax of the control statements and the conditions of their use.

Chapter 10. Creating JCL Using Control Statements

IBM File Export for z/OS uses control statements to specify information about the source data to be extracted and target data to be created by IBM File Export for z/OS. As you use the ISPF panels that comprise the IBM File Export for z/OS user interface, you are creating the associated control statements behind the scenes. It is not necessary for you to modify the file that is created. However, if you want to bypass the user interface and create a JCL file directly, you can do so using the control statement syntax as described in this chapter. Some parts of the IBM File Export for z/OS control statement syntax are similar to the SQL (Structured Query Language) definitions of SELECT and INSERT. For more information on extracting data using IBM File Export for z/OS without using the user interface, see the chapter "Working with the IBM File Export Batch Facility" in the *IBM File Export for z/OS: User's Guide*.

Control statement syntax

The basic structure of the IBM File Export for z/OS control statement syntax is described in this chapter. The general order of statements in a file of control statements is important. Because the parser for the control statement language is a one-pass parser, statements that define program constructs rely on statements defining higher-level constructs appearing ahead of them in the control statement flow. Additionally, statement order depends on the kind of export you are performing. Thus, for example, you must define a segment before defining the fields that comprise it, or define a database prior to defining the tables that comprise it.

Table 2 on page 204 lists each statement and its general position in the flow. However, the statement order depends on the kind of export that you are performing. As you define the control statement flow, you may omit statements, but you must maintain the statement order.

General order of control statements

Review the table cited for a detailed list of control statements in the order in which they should appear. In general, the order is:

- OPTIONS statement
- AUDIT
- EXITNAME
- EXITACTIVE
- Global field and literal specifications
- Extract type hierarchy (DB2 versus IMS)

IMS example

This example lists the order of statements used when defining an IMS export:

- IMSID(imsid);
- <IMSOPTIONS> statements
- SOURCEDBD
- TARGETDBD

- SEGMENT
- Segment fields
- Type specification
- Type fields
- [Resultset specification]
- Select specification
- [SOURCEDBD] <==only if the output will be FEUNLOAD
- SEGMENT
- Segment fields
- Type specification
- Type fields
- [Resultset specification]
- Select specification

Names used in control statements

ABXUTIL supports COBOL-like names -- that is, a hyphen may appear within a name. If a name contains a hyphen, it must follow the rules for COBOL names. As a consequence, where you have intended a minus sign (-), you must surround the minus sign with blanks except in the case where you are using the minus sign to express date deltas or negative numbers. For example, -14 to express a negative 14 is correct. Here is another correct example:

```
date(dtype1) = char 'yyyy/mm/dd' deltas (year-1, month-1, day-1);
```

Applicability of control statements based on type of source data

The following table maps the control statements against the type of source data to which they apply. The last column of the table provides a reference to the write-up about this control statement in the reference section of this Appendix.

Table 2. Control Statements Mapped to Source Data Type

Control Statements (listed in the order in which they <i>must</i> appear in the control statement flow)	DB2 Processing Mode	BMP/DBB/DLI /DEDB/IMS Processing Mode	VSAM or Sequential Processing Mode	See...
OPTIONS	X	X	X	Page 228
SOURCEDBD		X		Page 244
TARGETDBD		X		Page 248
AUDIT	X	X	X	Page 207
WUNIT	X	X	X	Page 256
EXITNAME			X	Page 216
EXITACTIVE	X	X	X	Page 216
IMSID		X		Page 220
PSBNAME		X		Page 233
INPCB#		X		Page 221

Table 2. Control Statements Mapped to Source Data Type

Control Statements (listed in the order in which they <i>must</i> appear in the control statement flow)	DB2 Processing Mode	BMP/DBB/DLI /DEDB/IMS Processing Mode	VSAM or Sequential Processing Mode	See...
OUTPCT#		X		Page 232
ALTRRC00		X		Page 207
DYNPSB		X		Page 215
DPSBPFX		X		Page 214
DPSBSFX		X		Page 215
DPSBMAC		X		Page 214
DOPTLIB		X		Page 214
RESLIB1		X		Page 236
RESLIB2		X		Page 236
RESLIB3		X		Page 236
DFSVSAMP		X		Page 213
LOCKMAX		X		Page 224
SSM		X		Page 245
AGN		X		Page 207
ALTID		X		Page 207
CPUTIME		X		Page 210
DIRCA		X		Page 213
NBA		X		Page 224
OBA		X		Page 225
OPT		X		Page 228
PARDLI		X		Page 232
STIMER		X		Page 245
BKO		X		Page 208
DLTDUPS		X		Page 213
BUF		X		Page 208
DBDLIB		X		Page 212
DBRC		X		Page 212
EXCPV		X		Page 216
FMTO		X		Page 219
GSGNAME		X		Page 220
IMSPLEX		X		Page 220
IRLM		X		Page 221
IRLMNAME		X		Page 222
MON		X		Page 224
PROCOPT		X		Page 232

Table 2. Control Statements Mapped to Source Data Type

Control Statements (listed in the order in which they <i>must</i> appear in the control statement flow)	DB2 Processing Mode	BMP/DBB/DLI /DEDB/IMS Processing Mode	VSAM or Sequential Processing Mode	See...
SWAP		X		Page 246
RESULTSET		X		Page 246
RELATIONSHIP (formerly SOURCEDD) CONTENT	X	X	X	Page 233
SOURCEDB2	X			Page 244
TARGETDB2	X			Page 248
DATE	X	X	X	Page 210
SEGMENT		X		Page 237
LAST_SEGMENT		X		Page 222
TABLE	X			Page 246
FIELD	X	X	X	Page 217
OFIELD	X	X	X	Page 225
TYPE		X	X	Page 251
SELECT	X	X	X	Page 239
EXCLUDE		X	X	Page 215
COLLECT		X	X	Page 208
INSERT		X (LOAD mode only)	X (LOAD mode only)	Page 221
THEN INSERT	X	X	X	Page 248
SKIP (global)			X	Page 244
SKIP (non-global)	X	X	X	Page 244
LIMIT (global)			X	Page 222
LIMIT (non--global)	X	X	X	Page 222
LITERAL		X	X	Page 222
VARIABLE	X	X	X	Page 254
COLUMN	X			Page 209

Control statement syntax, definitions, and descriptions

This section contains a description of each control statement and illustrates its syntax. Statements appear in this section in alphabetical order. There are some statements or statement parameters that require an explanation or elaboration beyond that in these descriptions. In that case, a reference is provided to a section of this book where you can obtain additional detailed information.

AGN

Use the AGN declarative to specify the Application Group Name to be used for interregion communication security. This parameter is passed to the IMS region controller in the appropriate position of the input parameters. This keyword is valid when BMP is specified on the OPTIONS keyword. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the AGN declarative. The syntax is as follows:

```
AGN(agn_name);
```

where:

agn_name

is a 1- to 8-byte Application Group Name to be used for interregion communication security.

If this declarative is specified in the JCL, it will override the AGN specification in the ABXOIMS0 options control block for this execution.

ALTID

Use the ALTID declarative to specify the IMS SSID that the IMS region will attempt to connect to if the IMS SSID specified on the IMSID parameter is unavailable. This parameter is passed to the IMS controller in the appropriate position of the input parameters. This keyword is valid when BMP is specified on the OPTIONS keyword. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the ALTID declarative. The syntax is as follows:

```
ALTID(ssid);
```

where:

ssid is a 1- to 4-byte name (or blanks).

If this declarative is specified in the JCL, it will override the ALTID specification in the ABXOIMS0 options control block for this execution.

ALTRRC00

Use the ALTRRC00 declarative to specify the name of the program that your installation uses as an alternate to the IMS region controller program DFSRRC00. This keyword is valid when DLI, DBB, or BMP is specified on the OPTIONS keyword. The syntax is as follows:

```
ALTRRC00(alt_pgmname);
```

where:

alt_pgmname

is a 1- to 8-byte program name.

If this declarative is specified in the control statement stream, it will override the ALTRRC00 specification in the ABXOIMS0 options control block for this execution.

AUDIT

Use the AUDIT declarative to specify whether or not an audit trail should be created using your system SMF data sets. This declarative is used in conjunction with the SMFNO declarative and is valid for all processing types. The syntax is as follows:

AUDIT(N|Y|F);

where:

- N** Turns off the audit trail process.
- Y** Causes the creation of an audit trail using SMF records. If the step is unable to produce the audit trail, a message is issued indicating the reason that the audit trail process has been terminated. The step continues processing, but without producing an audit trail.
- F** Forces the creation of an audit trail using SMF records. If the step is unable to produce an audit trail, the step terminates.

If this declarative is specified in the control statement stream, it attempts to override the AUDIT specification in the ABXOGBL0 options control block for this execution. If the ABXOGBL0 options control block has the value of **F** (FORCE), any values specified in the JCL are ignored.

BKO

Use the BKO declarative to specify whether or not a dynamic backout should occur when a pseudoabend occurs. This declarative is passed to the IMS region controller in the appropriate position of the input parameters. This keyword is valid when DLI or DBB is specified on the OPTIONS keyword. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the BKO declarative. The syntax is as follows:

BKO(N|Y);

where:

- N** Indicates that dynamic backout should not be attempted. This is the default.
- Y** Indicates that dynamic backout should be attempted.

If this declarative is specified in the control statement stream, it overrides the BKO specification in the ABXOIMS0 options control block for this execution.

BUF

Use the BUF declarative to specify the number of 1K blocks to be used in calculating the size of the OSAM buffer subpools. This parameter is passed to the IMS region controller in the appropriate position of the input parameters. This keyword is valid when DLI or DBB is specified on the OPTIONS keyword. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the BUF declarative. The syntax is as follows:

BUF(blocks);

where:

blocks

is a 1- to 3-digit (1- to 3-byte) value.

If this declarative is specified in the control statement stream, it will override the BUF specification in the ABXOIMS0 options control block for this execution.

COLLECT

Use the COLLECT declarative to capture field data that can later be used in INSERT statements, SELECT statements, or predicates. COLLECT must follow a SELECT or a THEN INSERT statement. The syntax is as follows:

```
COLLECT(collect_name) from field_name;
```

where:

collect_name

is the name you provide for the COLLECT statement that you are defining. This name can be used in SELECT and INSERT statements to call this COLLECT statement.

field_name

is the name of the field from which you want to collect data.

COLLECT can be used to transport data from a parent to a child.

Note: All predicates (if any) must precede the PRINT, UNIQUE, QUALIFIER, SPUFI, MINIMUM, MAXIMUM, TABLE, and QUALIFIER declaratives.

COLUMN

Use the COLUMN declarative to alter data in a fetched column. When loading data into a target table, you can replace data, add data to the data already in a column, scramble data, or omit one or more columns from being loaded into the target table. The syntax is as follows:

```
COLUMN(column-name) ORDINAL(nn) NULLABLE SOURCE = content-name (based-on-column-name) NEW OMIT SCRAMBLE date-name;
```

where ORDINAL, NULLABLE, SOURCE, NEW, OMIT, SCRAMBLE, and *date-name* are optional.

ORDINAL:

Use the ORDINAL declarative to identify a column that is unnamed or has a name that is not unique within the list of columns in the SELECT list. To accomplish this, you can refer to the column by its ordinal position in the SELECT list. This is needed whenever:

- The column is not named as in a literal in the SELECT list, or is not named by a scalar (like substring) or date functions, user-defined function, and so forth
- The target system's column name is to be different than the name of the source column
- The same column name appears in more than one table and is to be included in the output.

The syntax is as follows:

```
ORDINAL(nn)
```

where *nn* is the position of the column in the SELECT list. For example, to reference the third item in the SQLDA, you can use ORDINAL(3).

NULLABLE:

Use this declarative to indicate that the column can be assigned a value of null.

SOURCE:

The syntax is as follows:

```
SOURCE = content-name(key)
```

This syntax specifies that the data for this column will be overlaid with the data obtained by resolving the CONTENT expression.

If COLUMN-NAME is not in the SQLDA (in the SELECT list), IBM File Export assumes that you are adding a column. This requires that the column contain text. If the content text is present but not matched, the column is set to NULL if it is nullable; otherwise it is set to the zeroes or blanks as specified in the FILLER statement in the OPTIONS declarative.

NEW:

SOURCE = *content-name*(key) NEW

SOURCE, combined with NEW, adds a new (additional) column to the output data. The data type of the CONTENT must match defined data types valid for this file structure, with the exception of VARCHAR, which can be produced from CHAR.

OMIT:

Specifying OMIT causes this column to be omitted from the output. The column is included in the SELECT list for use in the IN(SOURCE-name) predicate or for functionality in the CONTENT expression.

date-name:

Specifying *date-name* adjusts the date value by the values specified in the date control statement identified by the *date-name*.

SCRAMBLE:

Specifying SCRAMBLE permits you to easily hide sensitive data content. It substitutes new values for record content, keeping the style of the current content. It is not a simple character replacement, but it is repeatable.

Note that scrambling any data that becomes a VSAM key can cause many CI and CA splits when loading the data, because the collating sequence in the scrambled text can change versus the original sequence. Also note that scrambling a forty byte CHAR field will not produce the same results as scrambling its parts. That is, scrambling the first twenty bytes, then scrambling the next twenty will not produce the same text as a single scrambling of the original forty bytes in one operation.

CPUTIME

Use the CPUTIME declarative to specify the maximum task time, in minutes, that the BMP region is allowed. This parameter is passed to the IMS region controller in the appropriate position of the input parameters. This keyword is valid when BMP is specified on the OPTIONS keyword. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the CPUTIME declarative. The syntax is as follows:

```
CPUTIME(time_limit);
```

where:

time_limit

is a 1- to 4-byte value from 0 through 1440.

If this declarative is specified in the control statement stream, it will override the CPUTIME specification in the ABXOIMS0 options control block for this execution.

DATE

Use the DATE declarative to create a date statement. Date statements are used to modify columns that contain dates, as, for example, changing dates in existing test

data. You can create as many date statements as necessary. Once a date statement is created, it can be applied to a field or a column in a table using the FIELD or COLUMN declaratives. Date statements define the data type, the order of the year, month, and day within any field using the definition, and the change that is to be applied to existing dates. Date statements can be one of two data types:

CHARACTER

A character date where the date statement defines the length of the components of the date and the relative position of each component. Each component is one byte of data. Additionally it can define the presence of an edit character.

PACKED

For a PACKED mask, you are defining the 'nibbles' or half bytes. A PACKED mask must be an even number of characters in length. Within the field, depending on the position of the data, there can be a leading zero and there can be a 'zone' as the last nibble. No editing characters are allowed in a PACKED mask.

The syntax for a date statement is as follows:

```
DATE (date-name) = data_type 'date-definition-mask' deltas (year year_change,month month_change , day day_change)
```

where:

date-name

Type a unique name for the date statement that you are creating. This is the name you can use in a FIELD statement to refer to the date statement.

data_type

Specify the data type for the date statement that you are creating. Type CHAR to specify CHARACTER data type.

date-definition-mask

Specify a mask that is to be applied to the date. The mask defines the order and number of characters for each component of the date. The acceptable syntax for the mask differs depending on the data type specified in the **Data Type** field.

CHAR

The mask for a date in the form yymmdd is: 'YMMDD'. This mask definition must be surrounded by single quotes.

Julian dates are specified like 'yyyy.ddd' or 'yy.ddd' (here the edit character is the period). If the descriptive mask has no month specification (mm) and a three byte day specification (ddd), the date is taken to be Julian. Masks like 'dddy', 'yyyyddd' or 'ddd.yyyy' are also valid.

PACKED

For a PACKED mask, you are defining the half bytes. A PACKED mask must be an even number of characters in length. Within the field, depending on the position of the data, there can be a leading zero. The last half-byte can be a 'zone'. No editing characters are allowed in a PACKED mask. The following examples are valid masks:

- '0yyyymmddz'
- 'nnddyyyy'
- 'yyyydddz'
- '0yyyyddd'
- '0yyddd'

year_change

Specify the changes to be applied to the year in the date when using this date statement. To indicate that the year should decrease by the specified amount, use a minus sign (-).

month_change

Specify the changes to be applied to the month in the date when using this date statement. To indicate that the month should decrease by the specified amount, use a minus sign (-). The month change is not specified for Julian dates.

day_change

Specify the changes to be applied to the day in the date when using this date statement. To indicate that the day should decrease by the specified amount, use a minus sign (-).

Here are some examples:

```
date(dtype1) = char 'yyyy/mm/dd' deltas (year - 1, month - 1, day -1);
date(dtype2) = char 'yyyy/mm/dd' deltas (year + 1, month + 1 , day + 1);
date (j1) = char 'yyyy.ddd' deltas (year -1 , day - 30);
date (j2) = char 'yyyy.ddd' deltas (year +1 , day+30);
```

The following example demonstrates how a date statement identified by the name *date-name* can be used in a FIELD declarative:

```
field(field-name) = (43,10,char) date-name;
```

The actual FIELD declarative might then be:

```
field(my_field) = (43,10,char) dtype1
```

DBDLIB

Use the DBDLIB declarative to specify the data set name of the partitioned data set that contains the DBDLIB members associated with the IMS subsystem. This declarative is valid when the DLI, DBB, or BMP processing option is specified. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the DBDLIB declarative. The syntax is as follows:

```
DBDLIB(dbdlib_name);
```

where:

dbdlib_name

is a 1-44 character name of the data set that contains the DBDLIB members associated with the IMS subsystem.

DBRC

Use the DBRC declarative to specify whether Database Recovery Control is to be used for this execution. This parameter is passed to the IMS region controller in the appropriate position of the input parameters. This declarative is valid when the DLI or DBB processing option is specified. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the DBRC

declarative. The syntax is as follows:

```
DBRC(N|Y|blank);
```

where:

N is a 1-byte value indicating that DBRC should not be used for this execution.

Y is a 1-byte value indicating that DBRC should be used for this execution.

blank

indicates that the DBRC= value specified in the IMSCTL macro during IMS system definition is to be used.

If this declarative is specified in the control statement stream, it will override the DBRC specification in the ABXOIMS0 options control block for this execution.

DFSVSAMP

Use the DFSVSAMP declarative to specify the name of a data set, the contents of which specify the size and number of buffers in a subpool. This declarative will cause the data set specified to be dynamically allocated using the DFSVSAMP DD, unless the DFSVSAMP DD is coded in the JCL. If the DFSVSAMP DD statement is coded in the JCL, this keyword and its parameter are ignored. This declarative is valid when DLI or DBB is specified on the OPTIONS keyword. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the DFSVSAMP declarative. The syntax is as follows:

```
DFSVSAMP(dataset_name);
```

where:

dataset_name

is a 1- to 44-byte data set name. This value may also include a member name, if appropriate.

If this declarative is specified in the control statement stream, it will override the DFSVSAMP specification in the ABXOIMS0 options control block for this execution. If the DFSVSAMP DD statement is coded in the JCL, the DD statement will override this keyword specification in the ABXOIMS0 options block and in the control statement stream.

DIRCA

Use the DIRCA declarative to specify the number of 1K blocks to be reserved for the dependent interregion communications area. This parameter is passed to the IMS region controller in the appropriate position of the input parameters. This keyword is valid when BMP is specified on the OPTIONS keyword. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the DIRCA declarative. The syntax is as follows:

```
DIRCA(value);
```

where:

value is a 1- to 3-digit (1- to 3-byte) value.

If this declarative is specified in the JCL, it will override the DIRCA specification in the ABXOIMS0 options control block for this execution.

DLTDUPS

Use the DLTDUPS declarative to specify whether ABXUTIL should automatically delete any duplicate segments that are found in an IMS database when writing to that database. Valid values are Y or YES, indicating that duplicate segments should be deleted, or the default value, N or NO, indicating that duplicate segments should not

be deleted. This parameter is only valid when specified along with the TYPE=ENTRY keyword. The default is blank. This declarative is valid when the DLI, DBB, or BMP processing option is specified. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the DLTDUPS declarative. The syntax is as follows:

```
DLTDUPS(N|Y);
```

where:

N is a 1-byte value indicating that duplicate segments should not be deleted.

Y is a 1-byte value indicating that duplicate segments should be deleted.

DOPTLIB

Use the DOPTLIB declarative to specify the name of an ACBLIB data set, used to contain a dynamically generated PSB when executing a BMP region. This data set cannot be the first data set in the ACBLIB DD concatenation that is specified in the IMS control region and DLI/SAS region (if applicable). This declarative is valid when BMP is specified on the OPTIONS keyword and the dynamic PSB feature is active for this execution. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the DOPTLIB declarative. The syntax is as follows:

```
DOPTLIB(dataset_name);
```

where:

dataset_name

is a 1- to 44-byte data set name.

If this declarative is specified in the control statement stream, it will override the DOPTLIB specification in the ABXOIMS0 options control block for this execution.

DPSBMAC

Use the DPSBMAC declarative to specify the name of a MACLIB data set, used to contain the IMS PSBGEN, PCB, and SENSEG assembler macros used to generate PSBs. Typically, this is the IMS.ADFSMAC data set, created during IMS installation. This declarative is valid when DLI, DBB, or BMP is specified on the OPTIONS keyword and the dynamic PSB feature is active for this execution. The syntax is as follows:

```
DPSBMAC(dataset_name);
```

where:

dataset_name

is a 1- to 44-byte data set name.

If this declarative is specified in the control statement stream, it will override the DPSBMAC specification in the ABXOIMS0 options control block for this execution.

DPSBPFX

Use the DPSBPFX declarative to specify a value to be used as the PSB name prefix. The value in this field will be used in conjunction with declarative DPSBSFX to create a PSB name when a PSB is to be dynamically generated. This declarative is valid when DLI, DBB, or BMP is specified on the OPTIONS keyword and the dynamic PSB feature is active for this execution. The syntax is as follows:

```
DPSBPFX(name_prefix);
```

where:

name_prefix

is a 1- to 6-byte character string that follows the rules for PDS member names.

If this declarative is specified in the control statement stream, it will override the DPSBPFX specification in the ABXOIMS0 options control block for this execution.

DPSBSFX

Use the DPSBSFX declarative to specify a value to be used as the PSB name SUFFIX. The value to be specified should be in the upper range of decimal values and is used in conjunction with declarative DPSBPFX to create a PSB name when a PSB is to be dynamically generated. This declarative is valid when DLI, DBB, or BMP is specified on the OPTIONS keyword and the dynamic PSB feature is active for this execution. The syntax is as follows:

```
DPSBSFX(name_suffix);
```

where:

name_suffix

is a 2 decimal digit string ranging from 00 through 99.

If this declarative is specified in the control statement stream, it will override the DPSBSFX specification in the ABXOIMS0 options control block for this execution.

DYNPSB

Use the DYNPSB declarative to specify whether the Dynamic PSB feature is active for this execution. This declarative is valid when DLI, DBB, or BMP is specified on the OPTIONS keyword. The syntax is as follows:

```
DYNPSB(N|Y);
```

where:

N is a 1-byte value indicating that the Dynamic PSB feature is not active for this execution.

Y is a 1-byte value indicating that the Dynamic PSB feature is active for this execution.

If this declarative is specified in the control statement stream, it will override the DYNPSB specification in the ABXOIMS0 options control block for this execution.

EXCLUDE

Use the EXCLUDE declarative to exclude constructs from processing. The syntax is:

```
EXCLUDE;
```

For example,

```
SEGMENT(child_name);
```

```
EXCLUDE;
```

excludes the segment named in the `SEGMENT(child_name);` statement from processing and effectively halts any processing that would have proceeded down that leg of a hierarchy.

There is a special TYPE named DEFAULT whose syntax is TYPE(DEFAULT). It must be the last TYPE definition in the list -- within the definitions or within the SEGMENT. This type name can be used more than once. At runtime, if no type is matched, the TYPE is set to: 'SELECT *'. To override this behavior, use the following statements following all of the other TYPE declaratives:

```
TYPE(DEFAULT) ;  
EXCLUDE;
```

See "TYPE" on page 251 for more information.

EXCPVR

Use the EXCPVR declarative to specify whether the OSAM buffer pools are to be page fixed. This declarative is valid when DLI or DBB is specified on the OPTIONS keyword. The syntax is as follows:

```
EXCPVR(0|1);
```

where:

- 0 is a 1-byte value indicating that the buffer pools should not be page fixed for this execution.
- 1 is a 1-byte value indicating that the buffer pools should be page fixed for this execution.

If this declarative is specified in the control statement stream, it will override the EXCPVR specification in the ABXOIMS0 options control block for this execution.

EXITACTIVE

Use the EXITACTIVE declarative to specify whether user-supplied I/O exits are to be called in this execution. This declarative is valid for all processing types. The syntax is as follows:

```
EXITACTIVE(N|Y);
```

where:

- N is a 1-byte value indicating that the I/O exit is not to be called for this execution.
- Y is a 1-byte value indicating that the I/O exit is to be called for this execution.

If this declarative is specified in the control statement stream, it will override the EXITACTIVE specification in the ABXOGBL0 options control block for this execution.

EXITNAME

Use the EXITNAME declarative to specify a member name that specifies the program to be used as the I/O exit. This declarative is valid when SEQ or VSAM is specified on the OPTIONS keyword. The syntax is as follows:

```
EXITNAME(exit_name);
```

where:

exit_name

is a 1- to 8-byte member name that specifies the program to be driven as the I/O exit for this execution.

If this declarative is specified in the control statement stream, it will override the EXITNAME specification in the ABXOGBLO options control block for this execution.

FIELD

Use the FIELD declarative to define an input record beginning with the first data byte. On output, IBM File Export for z/OS will develop the required lengths and insert them in the output record as needed. (See also the control statement syntax for OFIELD on page 225). The scope of a field may be local or global. The syntax of the FIELD control statement is as follows:

```
FIELD(field_name) = (position|CALC,length,type) CALC as algorithm OCCURS(n)  
DEPENDING ON field_name PRECISION(nn) SCALE(nn) SIGNED (nn) SCRAMBLE date-name;
```

For complex group items, this group's length (TGLEN) and maximum length of a group (MAXLEN) can be included as follows:

```
FIELD(field_name) = (position|CALC,length,type) CALC as algorithm OCCURS(n)  
DEPENDING ON field_name PRECISION(nn) SCALE(nn) SIGNED (nn) TGLEN(CALC as algorithm)  
MAXLEN(nn)SCRAMBLE date-name;
```

On the FIELD statement, all of the parameters are optional with the exception of ***field_name***, ***length***, and ***type***.

field_name

The name of the field. *field_name* must be unique within scope of the IBM File Export for z/OS control statement definitions.

position|*CALC*

The position of the field. POSITION can be specified:

- As a numeric integer type constant like 1 or 100
- Relative to the end of the record like 'end - 10' (ten bytes from the end of the record). If the OPTIONS(OFFSETS) declarative is used, this value is an offset.
- Using a hex constant (x'22' or h'22') which will always be interpreted as an offset (never as a 'position').
- As a cumulative position/offset definition. For example:
 - Use the NEXT declarative to specify that the position is the next character after the prior field.
 - Use the NEXTC declarative to specify that the position is the next character after the prior field.
 - Use the NEXTI declarative to specify that the position is the next integer position after the prior field
 - Use the NEXTW declarative to specify that the position is the next word position after the prior field
 - Use the NEXTH declarative to specify that the position is the next halfword location after the prior field.
 - Use the NEXTD declarative to specify that the position is the next double word location after the prior field.

Information: NEXTx does NOT set the field attribute (WORD/DOUBLE) just the position; the first field can be defined with NEXTx and it is in position 1.

The appearance of the word **CALC** in the control statement syntax indicates that an algorithm or expression to be resolved follows. **CALC** is used in place of ***position*** when position and length must be calculated, as when the position of data to the right of the first instance of a variable length field has

no fixed position. The position can be expressed as a simple calculation of a constant (the fixed length fields preceding it) and an "occurs depending on" value (see "Understanding Occurs Depending On" on page 262) multiplied by a constant. Up to eight instances of an ODO multiplied by a constant are permitted. The number of times the ODO is needed is the subscripting depth. ODOs can be nested. If there is more than one variable length field to the left of a given field, for example, the calculation could be expressed as:

```
position=fixed-constant + constant1 * ODO1 * ODO2 * ODO3 + constant2 * ODO4 * ODO5...
```

length

The length of the field. LENGTH is numeric literal. It is the length of the bytes required in memory for the field. The length can also be specified as '*' (an asterisk) which means the rest of the record as seen at run time. The sum of offset and length cannot exceed the record length.

type The data type of the field. (For a description of these data types, see Table 3 on page 257.) You can specify any of the following types:

- CHAR
- INT or INTEGER
- SHORT
- HEX
- PACKED
- FLOAT
- DOUBLE

PRECISION

The precision of the numeric data.

SCALE

The scale of the numeric field.

SIGNED

Signifies that the numeric data is signed.

TGLEN

The length of this group. For complex group items, TGLEN may be expressed as an algorithm (CALC as algorithm).

MAXLEN

The maximum length of the group.

SCRAMBLE

Specifying SCRAMBLE permits you to easily hide sensitive data content. It substitutes new values for record content, keeping the style of the current content. It is not a simple character replacement, but it is repeatable.

Note that scrambling any data that becomes a VSAM key can cause many CI and CA splits when loading the data, because the collating sequence in the scrambled text can change versus the original sequence. Also note that scrambling a forty byte CHAR field will not produce the same results as scrambling its parts. That is, scrambling the first twenty bytes, then scrambling the next twenty will not produce the same text as a single scrambling of the original forty bytes in one operation.

date-name

The name of the (previously created) date alteration statement that is to be applied to this field. This declarative is optional. The date-name must have been previously defined in a DATE definition. For more information on DATE alteration statements see "DATE" on page 210.

Examples

Following are some valid examples for FIELD and OFIELD definitions:

```
ofield(of1) = (1,2,short);
ofield(filler) = (nextc,1,char);
ofield(of2a) = (5,5,char);
ofield(filler) = (nextc,7,char);
ofield(of3) = (50,4,integer) occurs (10) depending on of1 precision(9)
              scale(3) signed tglen(16);
ofield(of3a) = (50,4,integer) occurs (10) depending on of1 precision(9)
              scale(3) signed tglen(calc as 4 *of1 + 2 *of1)
              maxlen(100);
ofield(of3b) = (50,4,integer) occurs (10) depending on of1 precision(9)
              scale(3) signed tglen(calc as 4 *of1 ) maxlen(100);
ofield(filler) = (nextc,1,char);
ofield(of4) = (calc,10,char) calc as 33 + of1 * 4;

ofield(of6) = (calc,10,char) calc as 33 + 4 * of1;
ofield(of7) = (calc,10,char) calc as 33 + 3 *of1;
ofield(of8) = (calc,10,char) calc as 33 + of1 * 4 * of1 *of1;
ofield(ofa) = (calc,10,char) calc as 33 + of1 * 4 * of1 *of1 + of1 *4;
ofield(ofb) = (calc,10,char) calc as 33 + of1 * 4 * of1 *of1 + of1 * 4 * of1;
ofield(ofc) = (calc,10,char) calc as 33 + of1 * 4 * of1 *of1 +
              of1 * 4 * of1 +
              of1 * 4 * of1 +
              of1 * 4 * of1 scramble;

ofield(filler) = (nextc,1,char);

field(field-1)=(1,4,char);
field(field2) = (4,1,char);
field(field3) = (1,*,char),scramble;

field(p3)      = (end - 5,5,char);

field(l1)      = (1,2,char);
field(zz)      = (3,2,char);

field(count)   = (6,2,char);
field(fld11)   = (11,10,char);
field(repeater) = (21,10,char) occurs (3) depending on count
                  tglen(calc as 10 *count) maxlen(30);
field(fld51)   = (calc,10,char) calc as 21 + 10 * count + 15 * count-1 * count-2;
field(fld61)   = (calc,10,char) calc as 31 + 10 * count;

ofield(l1)     = (1,2,char);
ofield(zz)     = (3,2,char);
ofield(count)  = (6,2,char);

ofield(repeater) = (21,10,char) occurs (3) depending on count
                  tglen(calc as 10 *count) maxlen(30);
ofield(fld51)   = (calc,10,char) calc as 21 + 10 * count;
```

FMTO

Use the FMTO declarative to specify the type of IMS dump output that should be produced. This declarative is valid when DLI or DBB is specified on the OPTIONS keyword. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for a list of valid options and a description of their effect. The syntax is as follows:

```
FMTO(dump_option);
```

where:

dump_option

is a 1-byte character that indicates any IMS dump formatting option that should be in effect for this execution.

If this declarative is specified in the control statement stream, it will override the FMTO specification in the ABXOIMS0 options control block for this execution.

GSGNAME

Use the GSGNAME declarative to specify the Global Services Group name to be used for the RSR complex. This keyword is valid when DLI or DBB is specified on the OPTIONS keyword. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the GSGNAME declarative. The syntax is as follows:

```
GSGNAME(gsg_name);
```

where:

gsg_name

is a 1- to 8-byte name that identifies the Global Services Group this execution should connect to.

If this declarative is specified in the JCL, it will override the GSGNAME specification in the ABXOIMS0 options control block for this execution.

IMSID

Use the IMSID declarative to:

- Identify a specific set of default options contained in the ABXOIMS0 control block
- Identify which IMS subsystem a BMP region should connect to.

This keyword is valid when DLI, DBB, or BMP is specified on the OPTIONS keyword. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for a list of restrictions and considerations regarding the use of the IMSID keyword. The syntax is as follows:

```
IMSID(ims_ssid);
```

where:

ims_ssid

is a 1- to 4-byte name that indicates which entry in the ABXOIMS0 option control block should be used to populate any default processing options. It will also be used by a BMP region to determine which active IMS subsystem it should connect to.

IMSPLEX

Use the IMSPLEX declarative to specify the 5-byte character value to be passed to the SCI registration exit, DSPSCIX0. This keyword is valid when DLI or DBB is specified on the OPTIONS keyword. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for a description and details of the use of the IMSPLEX declarative. The syntax is as follows:

```
IMSPLEX(plex_name);
```

where:

plex_name

is a 5 character (5-byte) name.

If this declarative is specified in the JCL, it will override the IMSPLEX specification in the ABXOIMS0 options control block for this execution.

INPCB#

Use the INPCB# declarative when using static PSBs to indicate which PSB PCB should be used to READ the source database. This keyword is valid when DLI, DBB, or BMP is specified on the OPTIONS keyword. The syntax is as follows:

```
INPCB#(pcb_number);
```

where:

pcb_number

is a 1- to 4-byte number with a maximum value of 2400 that must be specified when using static PSBs.

This keyword is not supported in the ABXOGBL0 or ABXOIMS0 option control blocks.

INSERT

INSERT is valid in LOAD mode only. There is no input file for an INSERT. When working with IMS databases, objects must be segments in the TARGETDBD. The INSERT statement takes the form

```
INSERT INTO insert_predicate;
```

The INSERT predicate can include only literals, named literals, and variables.

To understand the differences between INSERT and THEN INSERT, see “Understanding INSERT and THEN INSERT” on page 262.

Because the records you are inserting may be variable length records, you should understand the use of slack bytes as described in “Working with variable length records” on page 261.

IRLM

Use the IRLM declarative to specify whether IRLM should be used for this execution. This declarative is valid when DLI or DBB is specified on the OPTIONS keyword. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for a description and details of the use of the IRLM declarative. The syntax is as follows:

```
IRLM(Y|N|blank);
```

where:

Y is a 1-byte value indicating that IRLM is to be used for this execution.

N is a 1-byte value indicating that IRLM is not to be used for this execution.

blank

indicates that the IRLM= value specified in the IMSCTL macro during IMS system definition is to be used.

If this declarative is specified in the control statement stream, it will override the IRLM specification in the ABXOIMS0 options control block for this execution.

IRLMNAME

Use the IRLMNAME declarative to specify the MVS subsystem name assigned to the IRLM that you want this execution to use. This keyword is valid when DLI or DBB is specified on the OPTIONS keyword. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for a description and details of the use of the IMSPLEX declarative. The syntax is as follows:

```
IRLMNM(irlm_name);
```

where:

irlm_name

is a 1- to 4-byte IRLM subsystem name that identifies the IRLM subsystem that this execution should use as the lock manager.

If this declarative is specified in the JCL, it will override the IRLMNM specification in the ABXOIMS0 options control block for this execution.

LAST_SEGMENT

LAST_SEGMENT applies to IMS only. Use the LAST_SEGMENT declarative to specify the name of the segment that is to be the last segment processed in the hierarchy. It is a signal that processing is not to proceed further down the hierarchy; that is, it excludes from processing any children of this parent. The syntax is as follows:

```
LAST_SEGMENT(segment-name)
```

LIMIT

Use the LIMIT declarative to specify the number of records to be produced of a given SEGMENT or TYPE. The syntax is as follows:

```
LIMIT(nn) {THEN ONE IN mmm;
```

where

nn Specifies how many of a given SEGMENT or TYPE are to be produced. When the limit is met, the optional THEN ONE IN *mmm* takes effect.

THEN ONE IN *mmm*

Is optional and specifies that, after the limit is reached, then one in every *mmm* records is also produced. For IMS, SKIP, LIMIT, and THEN ONE IN are recycled each time the segment level is entered. For sequential and VSAM data, the LIMIT value is global and not reset, as no logical traversing of the data occurs.

See also "Limiting output record counts using SKIP and LIMIT" on page 258.

LITERAL

Use the LITERAL declarative to define a named literal. A LITERAL is a constant. There are two basic literal formats

- Implicit literal - such as 'TX' as found in a statement like WHERE STATE = 'TX'
- Named literals - a literal is defined once and assigned a name. You can refer to the literal by name rather than typing the value each time you need to use it.

For example, if logic dictates that states be tested in several expressions looking for the state of Texas, you can define a named literal and not have to enter it in the text repeatedly (this is advantageous for long literals). Literals are taken as their native length and type. Literals are sized to match the length of the corresponding field in the predicate. The syntax is as follows:

```
LITERAL (literal_name) = constant (length,data_type);
```

where *length* and *data_type* are optional and:

literal_name

The name for the literal that you are defining. The name must be unique within the IBM File Export for z/OS control statement syntax.

length

The length of the literal, which is the length in bytes required in memory for the field. Length is a numeric constant of type integer. The length can also be specified as '*' (an asterisk) which means the rest of the record as seen at run time. The sum of offset and length cannot exceed the record length. This declarative is optional.

data_type

The data type of the field. (For a description of these data types, see Table 3 on page 257.) You can specify any of the following types:

- CHAR
- INT or INTEGER
- SHORT
- HEX
- PACKED
- FLOAT
- DOUBLE

This declarative is optional.

There are many types of literals such as:

- An integer type literal like 1 or 10 or -14
- A floating point type of constant like 123.456e+10
- A hex constant like X'F0F1' or H'F0F1'
- A packed literal like p'-123'
- A character literal like "The customer"s name'

Examples:

- LITERAL(LIT1) = 1;
Produces an INTEGER.
- LITERAL(LIT1) = 1 (2,SHORT);
Produces a SHORT.
- LITERAL(LITCHAR) = 'X' (10,CHAR);
Produces a ten byte left justified literal.
- LITERAL(LIT1) = 'x';

Produces self defining literal syntax (where the data type and length are implicit). This example has an implicit length of one and an implicit data type of CHAR.

A literal must be defined before it is used. Character literals may be delineated with either the apostrophe or the double quote. If a literal is started with a quote, it must end in a quote. DBCS literals must be delineated with the shift-out (14) and shift-in (15).

LOCKMAX

Use the LOCKMAX declarative to override the PSBGEN LOCKMAX value (if one is specified) for this execution. This keyword is valid when DLI, DBB, or BMP is specified on the OPTIONS keyword. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the LOCKMAX declarative. The syntax is as follows:

```
LOCKMAX(value);
```

where:

value is a 1- to 5-digit (1- to 5-byte) value.

If this declarative is specified in the control statement stream, it will override the LOCKMAX specification in the ABXOIMS0 options control block for this execution.

MON

Use the MON declarative to specify whether the IMS monitor should be active for this execution. This declarative is valid when DLI or DBB is specified on the OPTIONS keyword. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for a description and details of the use of the IRLM declarative. The syntax is as follows:

```
MON(Y|N);
```

where:

Y is a 1-byte value indicating that the IMS monitor is to be used for this execution.

N is a 1-byte value indicating that the IMS monitor is not to be used for this execution.

If this declarative is specified in the control statement stream, it will override the MON specification in the ABXOIMS0 options control block for this execution.

NBA

Use the NBA declarative to specify the number of Fast Path database buffers that are to be made available for this execution. This keyword is valid when BMP is specified on the OPTIONS keyword. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the NBA declarative. The syntax is as follows:

```
NBA(value);
```

where:

value is a 1- to 5-digit (1- to 5-byte) number.

If this declarative is specified in the control statement stream, it will override the NBA specification in the ABXOIMS0 options control block for this execution.

OBA

Use the OBA declarative to specify the number of additional page-fixed Fast Path database buffers that are to be made available for this execution if the normal allotment has been used. This keyword is valid when BMP is specified on the OPTIONS keyword. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the OBA declarative. The syntax is as follows:

```
OBA(value);
```

where:

value is a 1- to 4-digit (1- to 4-byte) number.

If this declarative is specified in the control statement stream, it will override the OBA specification in the ABXOIMS0 options control block for this execution.

OFIELD

Use the OFIELD declarative to permit duplicate names in an input copybook definition and an output copybook definition. OFIELD also permits you to define an output record beginning with the first data byte. (See also the control statement syntax for FIELD on page 217.) The scope of a field may be local or global.

The syntax for OFIELD is the same as that for FIELD, but is only allowed in a STRICT-COLUMN-LIST. (While a column list (*column1, column2, column3,...*) may contain field names, collects, literals, and so forth, a strict column list may contain only OFIELD names.) The rules of scope apply to OFIELDS. That is, the scope of an OFIELD may be local or global. The syntax for OFIELD is:

```
OFIELD(field_name) = (position|CALC,length,type) CALC as algorithm OCCURS(n)  
DEPENDING ON field_name PRECISION(nn) SCALE(nn) SIGNED (nn) SCRAMBLE date-name;
```

For complex group items, this group's length (TGLEN) and maximum length of a group (MAXLEN) can be included as follows:

```
OFIELD(field_name) = (position|CALC,length,type) CALC as algorithm OCCURS(n)  
DEPENDING ON field_name PRECISION(nn) SCALE(nn) SIGNED(nn) TGLEN(CALC as algorithm)  
MAXLEN(nn)SCRAMBLE date-name;
```

On the OFIELD statement, all of the parameters are optional with the exception of ***field_name***, ***length***, and ***type***.

field_name

The name of the ofield. *field_name* must be unique within scope of the IBM File Export for z/OS control statement definitions.

position | ***CALC***

The position of the field. POSITION can be specified:

- As a numeric integer type constant like 1 or 100
- Relative to the end of the record like 'end - 10' (ten bytes from the end of the record). If the OPTIONS(OFFSETS) declarative is used, this value is an offset.
- Using a hex constant (x'22' or h'22') which will always be interpreted as an offset (never as a 'position').
- As a cumulative position/offset definition. For example:
 - Use the NEXT declarative to specify that the position is the next character after the prior field.

- Use the NEXTC declarative to specify that the position is the next character after the prior field.
- Use the NEXTI declarative to specify that the position is the next integer position after the prior field
- Use the NEXTW declarative to specify that the position is the next word position after the prior field
- Use the NEXTH declarative to specify that the position is the next halfword location after the prior field.
- Use the NEXTD declarative to specify that the position is the next double word location after the prior field.

Note: NEXTx does NOT set the field attribute (WORD/DOUBLE), just the position. The first field can be defined with NEXTx and it is in position 1.

The appearance of the word **CALC** in the control statement syntax indicates that an algorithm or expression to be resolved follows. **CALC** is used in place of *position* when position and length must be calculated, as when the position of data to the right of the first instance of a variable length field has no fixed position. The position can be expressed as a simple calculation of a constant (the fixed length fields preceding it) and an "occurs depending on" value (see "Understanding Occurs Depending On" on page 262) multiplied by a constant. Up to eight instances of an ODO multiplied by a constant are permitted. The number of times the ODO is needed is the subscripting depth. ODOs can be nested. If there is more than one variable length field to the left of a given field, for example, the calculation could be expressed as:

```
position=fixed-constant + constant1 * 0D01 * 0D02 * 0D03 + constant2 * 0D04 * 0D05...
```

length

The length of the field. LENGTH is numeric literal. It is the length of the bytes required in memory for the field. The length can also be specified as '*' (an asterisk) which means the rest of the record as seen at run time. The sum of offset and length cannot exceed the record length.

type

The data type of the field. (For a description of these data types, see Table 3 on page 257.) You can specify any of the following types:

- CHAR
- INT or INTEGER
- SHORT
- HEX
- PACKED
- FLOAT
- DOUBLE

PRECISION

The precision of the numeric data.

SCALE

The scale of the numeric field.

SIGNED

Signifies that the numeric data is signed.

TGLEN

The length of this group. For complex group items, TGLEN may be expressed as an algorithm (CALC as algorithm).

MAXLEN

The maximum length of the group.

SCRAMBLE

Specifying SCRAMBLE permits you to easily hide sensitive data content. It substitutes new values for record content, keeping the style of the current content. It is not a simple character replacement, but it is repeatable.

Note that scrambling any data that becomes a VSAM key can cause many CI and CA splits when loading the data, because the collating sequence in the scrambled text can change versus the original sequence. Also note that scrambling a forty byte CHAR field will not produce the same results as scrambling its parts. That is, scrambling the first twenty bytes, then scrambling the next twenty will not produce the same text as a single scrambling of the original forty bytes in one operation.

date-name

The name of the (previously created) date alteration statement that is to be applied to this field. This declarative is optional. The date-name must have been previously defined in a DATE definition. For more information on DATE alteration statements see "DATE" on page 210.

Examples

Following are some valid examples for FIELD and OFIELD definitions:

```
ofield(of1) = (1,2,short);
ofield(filler) = (nextc,1,char);
ofield(of2a) = (5,5,char);
ofield(filler) = (nextc,7,char);
ofield(of3) = (50,4,integer) occurs (10) depending on of1 precision(9)
              scale(3) signed tglen(16);
ofield(of3a) = (50,4,integer) occurs (10) depending on of1 precision(9)
              scale(3) signed tglen(calc as 4 *of1 + 2 *of1)
              maxlen(100);
ofield(of3b) = (50,4,integer) occurs (10) depending on of1 precision(9)
              scale(3) signed tglen(calc as 4 *of1 ) maxlen(100);
ofield(filler) = (nextc,1,char);
ofield(of4) = (calc,10,char) calc as 33 + of1 * 4;

ofield(of6) = (calc,10,char) calc as 33 + 4 * of1;
ofield(of7) = (calc,10,char) calc as 33 + 3 *of1;
ofield(of8) = (calc,10,char) calc as 33 + of1 * 4 * of1 *of1;
ofield(ofa) = (calc,10,char) calc as 33 + of1 * 4 * of1 *of1 + of1 *4;
ofield(ofb) = (calc,10,char) calc as 33 + of1 * 4 * of1 *of1 + of1 * 4 * of1;
ofield(ofc) = (calc,10,char) calc as 33 + of1 * 4 * of1 *of1 +
              of1 * 4 * of1 +
              of1 * 4 * of1 +
              of1 * 4 * of1 scramble;

ofield(filler) = (nextc,1,char);

field(field-1)=(1,4,char);
  field(field2) = (4,1,char);
field(field3) = (1,*,char),scramble;

field(p3)      = (end - 5,5,char);

field(ll)      = (1,2,char);
field(zz)      = (3,2,char);
```

```

field(count)      = (6,2,char);
field(fld11)     = (11,10,char);
field(repeater)  = (21,10,char) occurs (3) depending on count
                  tglen(calc as 10 *count) maxlen(30);
field(fld51)     = (calc,10,char) calc as 21 + 10 * count + 15 * count-1 * count-2;
field(fld61)     = (calc,10,char) calc as 31 + 10 * count;

ofield(11)       = (1,2,char);
ofield(zz)       = (3,2,char);
ofield(count)    = (6,2,char);

ofield(repeater) = (21,10,char) occurs (3) depending on count
                  tglen(calc as 10 *count) maxlen(30);
ofield(fld51)    = (calc,10,char) calc as 21 + 10 * count;

```

OPT

Use the OPT declarative to specify what action the dependent region should take if the IMS subsystems named using the IMSID or ALTID are not available. This declarative is valid when BMP is specified on the OPTIONS keyword. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for a description and details of the use of the IRLM declarative. The syntax is as follows:

```
OPT(W|N|C);
```

where:

- W** is a 1-character value indicating that the region should WAIT for the control program to start.
- N** is a 1-character value indicating that the region should issue a WTOR to ask the operator what should be done.
- C** is a 1-character value indicating that the region should be cancelled.

If this declarative is specified in the control statement stream, it will override the OPT specification in the ABXOIMS0 options control block for this execution.

OPTIONS

Use the OPTIONS declarative to specify the global options for IBM File Export for z/OS. Place the OPTIONS declarative before all other declaratives except the print command.

```

OPTIONS(VSAM|SEQ|DB2|IMS|DEDB|DLI|DBB|BMP|FEUNLOAD),PLAN(plan_name),NOCONTROL,EDITON
LY,DECIMAL|COMMA,FILLER(BLANK|BLANKS,ZERO|ZEROS|x'00'...x'FF'),STAR,FIXED(nn),OFFSE
TS,QUIET|NOQUIET,QUALIFIER(table_owner),FIELDS|NOFIELDS,GROUP,DUMMY,LOAD);

```

Data type DB2|VSAM|SEQ|IMS:

You must specify the data type of the data that you are going to export before any other OPTIONS declaratives. You can specify:

- VSAM
- SEQ
- DB2
- IMS
- DEDB
- DLI
- DBB

- BMP
- FEUNLOAD

You can also specify limited combinations of input and output types:

- You can use sequential input and produce VSAM output:
`OPTIONS((SEQ,VSAM),...`
- You can use VSAM input and produce sequential output:
`OPTIONS((VSAM,SEQ),...`
- You cannot use SEQ or VSAM as input to an IMS output format.
- You can use read files in either of the IMS UNLOAD formats (IMS or DEDB) and produce an output file in either of the IMS UNLOAD formats.
- You can use one of the IMS region types (DLI, DBB, or BMP) to write the data directly to an IMS database:
`OPTIONS((IMS,BMP),...`
- You can read an IMS database using one of the IMS region types (DLI, DBB, or BMP) and produce an output file in either of the IMS UNLOAD formats (IMS or DEDB):
`OPTIONS((DLI,DEDB),...`
- You cannot use any of the IMS formats as input to create a SEQ or VSAM output data set.
- You can specify a single data type to indicate that the input and output formats will be the same :
`OPTIONS(SEQ,...`
or, similarly:
`OPTIONS((SEQ,SEQ),...`
- FEUNLOAD is only valid if the input IMS region type is DLI, DBB, or BMP. Specifying FEUNLOAD indicates that Direct IMS processing will be performed using random access and a File Export UNLOAD file is to be produced. Once you have produced the File Export UNLOAD file, you can insert the selected records into an IMS database using program ABXFELD0. Information about this program is provided in the section "Inserting data into IMS databases using the File Export UNLOAD file" on page 259.

The other declaratives available in the OPTIONS statement vary depending on the type of data specified.

PLAN(plan_name):

Use the PLAN(plan_name) declarative to specify that ABXUTIL can operate using a plan other than its own default plan name (ABXUTIL). Specifying a plan name is useful when you want to have different versions of ABXUTIL in your system that require different plans, such as the case where a plan might correspond to the version level.

NOCONTROL:

Use the NOCONTROL declarative to specify that, when extracting from a DB2 source, no control file is present because the user interface was not used to create this control statement syntax. It also indicates that no control information for the DB2 load utility will be generated.

EDITONLY:

Use the EDITONLY declarative to specify that RUN will not be invoked, only edit the command syntax. This declarative is used when working with the user interface to check the syntax of the IBM File Export for z/OS control statement syntax that has been created.

Decimal character DECIMAL | COMMA:

Specify either DECIMAL or COMMA to determine which character will be used in real numbers. DECIMAL is the default but can be replaced by COMMA for European floating point (real) numbers. If you specify the comma as the decimal point, the contained parser will not be aware of syntax position. For example, when in the 'decimal = comma' mode, commas juxtaposed to numbers are taken as the decimal point. In: `literal(literal_name) = 2,3 (4,float);` will parse the 2,3 as the equivalent of 2.3 but the '4,float' is taken as '4.float' which is not recognized as valid syntax. To use the comma as the decimal point, separate numbers (that are not real numbers) from adjacent commas with a blank. In the above example, write: `literal(literal_name) = 2,3 (4 ,float);` which separates the comma following the numeral 4 and allows the parser to correctly identify the numeral 4 as an integer, not as a real number.

LOAD:

Use the LOAD declarative to specify that you want to run in LOAD mode. LOAD is a mode of processing where only inserts with literals are executed. You can specify:

LOAD

or

LOAD((*nnn*) NOECHO,NOINFO)

where

- *nnn* is the number of times that you want the system to execute your list of INSERT statements. This is especially useful if you are using VARIABLES as you can use this to create many unique records or segments.
- NOECHO suppresses echoing the ABXCTL stream to the print file
- NOINFO suppresses emitting the informational statements associated with parsing the INSERT statements to prevent your SYSOUT from becoming overloaded

For DB2 only, the LOAD declarative can have the format

LOAD(resume|replace|reuse)

where

- resume indicates whether records are to be loaded into an empty or non-empty table space.
- replace indicates whether the table space and its indexes need to be reset to empty before records are loaded.
- reuse is used with replace and specifies that LOAD should logically reset and reuse DB2-managed data sets without deleting and redefining them.

FILLER:

FILLER(BLANK|BLANKS,ZERO|ZEROES|00...FF)

Use the FILLER declarative to specify what should be used as a slack byte filler or pad character. Use BLANK or BLANKS to specify that a blank should be used as the pad

character. Use *ZERO* or *ZEROES* to specify that '0' should be used as the pad character. Use one of the 1-byte hexadecimal values 00 through FF to indicate that the specified hexadecimal value should be used as the pad character.

Wild card character STAR | ASTERISK | PERCENT:

You must specify the character that you want to use as a wild card character. You can specify:

- STAR to specify that '*' is the wild card character.
- ASTERISK to specify that '*' is the wild card character.
- PERCENT to specify that '%' is the wild card character.

Note: In DB2 mode, wild card processing is native SQL and DB2 uses only the percent sign (%). This value cannot be overridden.

This is the multiple character wild card that is used in evaluating 'LIKE' predicates.

FIXED:

Use the FIXED declarative to force fixed length record output of *nn*. The syntax is as follows:

```
FIXEDnn
```

Normally this utility produces variable length record output, however, the output can be forced to fixed with an LRECL of *nn*. This directive is ignored for IMS processing.

QUIET | NOQUIET:

Use the QUIET declarative to suppress runtime messages normally emitted by PRINT requests on SELECT and INSERT statements and suppress the printing of records that fail the UNIQUE directive. Use the NOQUIET declarative to cause ABXUTIL to produce the print output suppressed by the QUIET declarative.

QUALIFIER:

You can set the default QUALIFIER (owner) for any SPUFI output if SPUFI output is requested. The syntax is as follows:

```
QUALIFIER(owner_name)
```

where *owner_name* is the name of the table owner. All QUALIFIER statements declared at the SEGMENT level are inherited at the TYPE level and subsequently at the SELECT level. If QUALIFIER is declared at the TYPE level, it is inherited at the SELECT level. This element is optional.

NOFIELDS | FIELDS:

Use the NOFIELDS declarative to suppress the messages that are produced when the fields from the source DBD are automatically defined in the input DBD when the DBDLIB definition is run. By default, a message is emitted for each field, specifying its attributes. When NOFIELDS is specified, the fields are defined but the associated messages are not printed. Use the FIELDS declarative to cause ABXUTIL to produce the print output suppressed by the NOFIELDS declarative.

OFFSETS:

Use the OFFSETS declarative to define FIELDS using the offset approach. By default, the location of input fields is defined in the context of position (where the first byte is position one). When you specify OFFSETS, '0' is the first position.

GROUP:

Use the GROUP declarative in IMS processing to specify that selection for a set of segments is based on that set of segments satisfying the FINAL SELECT. For more information on the FINAL declarative, see FINAL 241.

DUMMY:

Use the DUMMY declarative to specify that IBM File Export for z/OS should not write to ABXOUT. This can be useful for populating an in-list associated with related records when the objective is only the in-list values, not the output data set (ABXOUT).

OUTPCB#

Use the OUTPCB# declarative when using static PSBs to indicate which PSB PCB should be used to WRITE to the target database. This keyword is valid when DLI, DBB, or BMP is specified on the OPTIONS keyword. The syntax is as follows:

```
OUTPCB#(pcb_number);
```

where:

pcb_number

is a 1- to 4-byte number with a maximum value of 2400 that must be specified when using static PSBs.

This keyword is not supported in the ABXOGBL0 or ABXOIMS0 option control blocks.

PARDLI

Use the PARDLI declarative to specify the parallel DL/I option. This declarative is valid when BMP is specified on the OPTIONS keyword. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for a description and details of the use of the PARDLI declarative. The syntax is as follows:

```
PARDLI(0|1);
```

where:

0 is a 1-character value indicating that DL/I processing is to be performed in the BMP region.

1 is a 1-character value indicating that DL/I processing is to be performed in the control region.

If this declarative is specified in the control statement stream, it will override the PARDLI specification in the ABXOIMS0 options control block for this execution.

PROCOPT

Use the PROCOPT declarative to specify the IMS PROCOPT that should be used to read databases when using a dynamically built PSB. Valid options are G and GO. This keyword is valid when DLI, DBB, or BMP is specified on the OPTIONS keyword. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the PROCOPT parameter. The syntax is as follows:

```
PROCOPT(G|GO);
```

PSBNAME

Use the PSBNAME declarative when using static PSBs to indicate which PSB should be used to access the source and target databases. This keyword is valid when DLI, DBB, or BMP is specified on the OPTIONS keyword. If you specify the PSBNAME keyword, you must also specify the keywords INPCB# and OUTPCB#. The syntax is as follows:

```
PSBNAME(psb_name);
```

where:

psb_name

is a 1- to 8-byte name of the PSB which contains PCBs to appropriately access both the source and target databases.

This keyword is not supported in the ABXOGBL0 or ABXOIMS0 option control blocks.

RELATIONSHIP

Use the RELATIONSHIP declarative to define a source or multiple sources and to select the data that is in a specified location in the source and will be written to the target output. The source can be used to populate an INLIST (used in predicates in subsequent SELECT statements). Multiple sources can be defined from the data set that is specified by the RELATIONSHIP declarative.

The RELATIONSHIP statement, combined with THEN INSERT INTO DD:ddname tie together several ABXUTIL executions that occur as separate steps. (That is, these statements will define an application-defined relationship; see “Defining relationships in IBM File Export” on page 5.) In the first execution of ABXUTIL, information is captured and will be used in subsequent executions of ABXUTIL. The captured information is placed into the DD specified by INSERT INTO DD:. Data to be selected from the RELATIONSHIP data set is identified by the sourcename1, sourcename2 values.

The data identified as *sourcename'n'* serves as a criterion to select additional records and the CONTENT of the records that are selected (for example, STATE=TX) or not selected (for example, STATE not equal to TX) is used in a subsequent execution of ABXUTIL.

The CONTENT declarative, then, is used in a SELECT list or a THEN INSERT in the following way:

```
SELECT field1, field2, content1(field3) from ...
```

In this case, content_name was defined on a RELATIONSHIP declarative, and the named field (field3) is some content in the current record that has the same value as the key that you have defined in the CONTENT portion of a RELATIONSHIP statement. The key and the field (in this instance, field3) must have the same data type and length.

The RELATIONSHIP declarative must follow the OPTIONS and DBDNAME declaratives. It must precede any SEGMENT declarative, TYPE declarative, or SELECT statement. The sources that are defined by this declarative are global in scope, but can only be referenced in the IN or INLIST predicate. The content of the data set that is defined by the RELATIONSHIP statement must have been created by a THEN INSERT INTO DD:ddname statement in a previous run of IBM File Export for z/OS or can be an already-existing file.

RELATIONSHIP DD:*ddname* (source-name1 = (*position,length,type*) , source-name2 = (*position,length,type*) CONTENT content-name key = (*position,length,type*) data = (*position,length,type*) date = *date-name* SCRAMBLE, *content-name2* key = (*position,length,type*)

where, when specified for RELATIONSHIP:

ddname

The name of the RELATIONSHIP data set. Multiple sources can be defined from this data set.

position

The position of the source data within the RELATIONSHIP data set. POSITION can be specified:

- As a numeric integer type constant like 1 or 100
- Using a hex constant (x'22' or h'22') which will always be interpreted as an offset (never as a 'position').

length

The length of the field. LENGTH is numeric literal. It is the length of the bytes required in memory for the field. The length can also be specified as '*' (an asterisk) which means the rest of the record as seen at run time. The sum of offset and length cannot exceed the record length.

type

The data type of the field. (For a description of these data types, see Table 3 on page 257.) You can specify any of the following types:

- CHAR
- INT or INTEGER
- SHORT (a two byte integer)
- HEX
- PACKED
- FLOAT
- DOUBLE

You can specify multiple source names within one RELATIONSHIP statement by separating the *source-name* = (*position,length,type*) declarations with a comma:

RELATIONSHIP:*ddname* *source-name* = (*position,length,type*), *source-name2* = (*position,length,type*), *source-name3* = (*position,length,type*);

where, when specified for CONTENT:

content_name

The name of the CONTENT data set. Multiple content sources can be defined.

key=position

The position of the key within the CONTENT data set. POSITION can be specified:

- As a numeric integer type constant like 1 or 100
- Using a hex constant (x'22' or h'22') which will always be interpreted as an offset (never as a 'position').

key=length

The length of the key field in the CONTENT data set. LENGTH is numeric literal. It is the length of the bytes required in memory for the field. The length can also be specified as '*' (an asterisk) which means the rest of the record as seen at run time. The sum of offset and length cannot exceed the record length.

key=type

The data type of the field. (For a description of these data types, see Table 3 on page 257.) You can specify any of the following types:

- CHAR
- INT or INTEGER
- SHORT (a two byte integer)
- HEX
- PACKED
- FLOAT
- DOUBLE

data=position

The position of the data within the CONTENT data set. POSITION can be specified:

- As a numeric integer type constant like 1 or 100
- Using a hex constant (x'22' or h'22') which will always be interpreted as an offset (never as a 'position').

data=length

The length of the data field in the CONTENT data set. LENGTH is numeric literal. It is the length of the bytes required in memory for the field. The length can also be specified as '*' (an asterisk) which means the rest of the record as seen at run time. The sum of offset and length cannot exceed the record length.

data=type

The data type of the field. (For a description of these data types, see Table 3 on page 257.) You can specify any of the following types:

- CHAR
- INT or INTEGER
- SHORT (a two byte integer)
- HEX
- PACKED
- FLOAT
- DOUBLE

date-name

Specifying *date-name* adjusts the date value by the values specified in the date definition identified by the *date-name*.

SCRAMBLE

Specifying SCRAMBLE permits you to easily hide sensitive data content. It substitutes new values for record content, keeping the style of the current content. It is not a simple character replacement, but it is repeatable.

Note that scrambling any data that becomes a VSAM key can cause many CI and CA splits when loading the data, because the collating sequence in the scrambled text can change versus the original sequence. Also note that scrambling a forty byte CHAR field will not produce the same results as scrambling its parts. That is, scrambling the first twenty bytes, then scrambling the next twenty will not produce the same text as a single scrambling of the original forty bytes in one operation.

More information about

RESLIB1

Use the RESLIB1 declarative to specify the name of the IMS SVC data set that contains the IMS nucleus and required service modules. This is typically known as the RESLIB. This keyword is valid when DLI, DBB, or BMP is specified on the OPTIONS keyword. When used in a concatenation, this data set will be the first. The syntax is as follows:

```
RESLIB1(dataset_name);
```

where:

dataset_name

is a 1- to 44-byte data set name.

If this declarative is specified in the control statement stream, it will override the RESLIB1 specification in the ABXOIMS0 options control block for this execution.

RESLIB2

Use the RESLIB2 declarative to specify the name of the IMS data set that contains any IMS dynamic allocation members, also known as DFSMDA members. This keyword is valid when DLI, DBB, or BMP is specified on the OPTIONS keyword. When used in a concatenation, this data set will be the second. The syntax is as follows:

```
RESLIB2(dataset_name);
```

where:

dataset_name

is a 1- to 44-byte data set name.

If this declarative is specified in the control statement stream, it will override the RESLIB2 specification in the ABXOIMS0 options control block for this execution.

RESLIB3

Use the RESLIB3 declarative to specify the name of the IMS data set that contains any user-supplied programs or exits needed to access IMS databases. This keyword is valid when DLI, DBB, or BMP is specified on the OPTIONS keyword. When used in a concatenation, this data set will be the third. The syntax is as follows:

```
RESLIB3(dataset_name);
```

where:

dataset_name

is a 1- to 44-byte data set name.

If this declarative is specified in the control statement stream, it will override the RESLIB3 specification in the ABXOIMS0 options control block for this execution.

RESULTSET

Use the RESULTSET declarative to specify a result set name. There are two formats for this declarative. Format 1 can be specified for all SEQ, VSAM, and IMS processing options. Format 2 is only allowed when IMS random access with complex application relationships is being used. The syntax is as follows:

```
RESULTSET (result_set_name);
```

or

```
RESULTSET (short_name_1) CHILD OF short-name_2 WHERE short-name_1.fieldlist = short-name_2.fieldlist;
```

A RESULTSET *must* be declared within the SOURCEDBD/SEGMENT/TYPE definition when you are working with IMS random access with complex application relationships and you are making selections from multiple DBDs.

Random access using GET UNIQUE processing will be activated when all of the following conditions are met:

- The SELECT associated with the RESULTSET is for the root segment,
- The fields in the parent-child relationship for the child exactly match the keys defined in the IMS DBD for this database, and
- There is only one SELECT for the root segment in this database.

If these conditions are not met, the database will be processed sequentially and root segments matching the criteria will be exported.

Parent-child relationships

You can define parent-child relationships within the RESULTSET declarative. The syntax is as follows:

```
CHILD OF PARENT = short-name_2 WHERE short-name_1.field1[,field2[,...fieldn]] = short-name_2.field1[,field2[,...fieldn]];
```

Only the first field in the parent-child relationship is qualified, both in the parent field and in the child field.

The number of fields specified for the child and the parent must be the same and each child field must have the same length and data type as its corresponding parent field.

Performance Note: The parent-child relationships exist in memory. As a parent record is read and then selected, its key is preserved in memory until the associated child record has been processed. If the keys are long, or if there are many parent records, this may require a large amount of memory.

The RESULTSET declarative must be followed immediately by a SELECT declarative.

SEGMENT

Use the SEGMENT declarative to specify a segment that will be used. The syntax is as follows:

```
SEGMENT (segment name) MINIMUM(nn) MAXIMUM(nn) PRINT(nn) TABLE(table_name)  
SPUFI(nnnnnnnn|*) QUALIFIER(owner_name);
```

A SEGMENT must be declared within the SOURCEDBD definition when working with IMS data and within the TARGETDBD when the INSERT declarative is used. At the SEGMENT level you can specify MINIMUM, MAXIMUM, PRINT, SPUFI, and QUALIFIER.

There is no default SEGMENT directive. Children are selected if the parent is selected. If you want a particular child to be excluded you can use:

```
SEGMENT(child_name);  
  
EXCLUDE;
```

MINIMUM

Use the MINIMUM declarative to specify the minimum length of the record that is produced by the SELECT statement from this SEGMENT. The syntax is as follows:

MINIMUM(*nn*)

where *nn* is the minimum length of the record that is produced by the SELECT statement from this SEGMENT. Values that are shorter than the length specified will be padded out to the specified length using the value specified in the FILLER. All SELECT statements under a TYPE inherit the MINIMUM specified at the TYPE level. Specifying the MINIMUM at the SELECT level overrides the inherited value. A declaration of MINIMUM(0) negates the inherited value. If MINIMUM is declared at the SEGMENT level, it is inherited at the TYPE level and subsequently at the SELECT level. If MINIMUM is declared at the TYPE level, it is inherited at the SELECT level. In IMS mode it is recommended not to specify MINIMUM because the SEGMENT MINIMUM and MAXIMUM are taken from the definition found in the target (to) DBD and propagate to the TYPES and the SELECTS. However, you could specify this at the TYPE level (or SELECT level) if there is a required difference in the limits for a particular type of record. This element is optional. See also “Specifying MINIMUM, MAXIMUM, PRINT, or SPUFI” on page 262.

MAXIMUM

Use the MAXIMUM declarative to specify the maximum length of the record that is produced by the SELECT statement from this SEGMENT. The syntax is as follows:

MAXIMUM(*nn*)

where *nn* is the maximum length of the record that is produced by the SELECT statement from this SEGMENT. Records that are produced by the SELECT statement that are longer than the specified length, result in an error at runtime. All SELECT statements under a TYPE inherit the MAXIMUM specified at the TYPE level. Specifying the MAXIMUM at the SELECT level overrides the inherited value. A declaration of MAXIMUM(0) negates the inherited value and suppresses the edits or printing. If MAXIMUM is declared at the SEGMENT level, it is inherited at the TYPE level and subsequently at the SELECT level. If MAXIMUM is declared at the TYPE level, it is inherited at the SELECT level. In IMS mode it is recommended not to specify MAXIMUM because the SEGMENT MINIMUM and MAXIMUM are taken from the definition found in the target (to) DBD and propagate to the TYPES and the SELECTS. However, you could specify this at the TYPE level (or SELECT level) if there is a required difference in the limits for a particular type of record. See also “Specifying MINIMUM, MAXIMUM, PRINT, or SPUFI” on page 262.

PRINT

You can use PRINT to print the output returned by a SELECT statement. The syntax is:

PRINT(*nn*)

where *nn* is the number of records that you want to print. The default value for *nn* is 1. For example, PRINT(12) means print the first 12 records that are produced by the SELECT statement from this SEGMENT. All PRINT statements that are declared at the SEGMENT level are inherited at the TYPE level and subsequently at the SELECT level. If PRINT is declared at the TYPE level, it is inherited at the SELECT level. This element is optional. See also “Specifying MINIMUM, MAXIMUM, PRINT, or SPUFI” on page 262.

TABLE

TABLE is valid for DB2 only. Use the TABLE declarative to identify the source table (or tables) and the target table to be loaded into in the target DB2 system. The syntax is as follows:

```
TABLE(table_name)
```

where *table_name* is the name that identifies the table. A table named at the SEGMENT level is inherited at the TYPE level and subsequently at the SELECT level. If a table is named at the TYPE level, it is inherited at the SELECT level.

SPUFI

Use the SPUFI declarative to indicate that SPUFI formatted records (to the specified count) are to be produced. The syntax is as follows:

```
SPUFI(nnnnnnnn|*)
```

where *nnnnnnnnnn* represents the maximum number of records to be produced. The maximum value that can be specified is 2147483647. If asterisk (*) is specified, it is the same as specifying the maximum number, 2147483647. All SPUFI keywords declared at the SEGMENT level are inherited at the TYPE level and subsequently at the SELECT level. If a SPUFI keyword is declared at the TYPE level, it is inherited at the SELECT level. This element is optional, but if used, you must specify ABXSQL DD. See also "Specifying MINIMUM, MAXIMUM, PRINT, or SPUFI" on page 262.

QUALIFIER

You can use QUALIFIER to set the table qualifier and owner for any SPUFI output from the SELECT statement from this SEGMENT. The syntax is as follows:

```
QUALIFIER(owner_name)
```

where *owner_name* is the name of the table owner. All QUALIFIER statements declared at the SEGMENT level are inherited at the TYPE level and subsequently at the SELECT level. If QUALIFIER is declared at the TYPE level, it is inherited at the SELECT level. This element is optional.

SELECT statements

For SELECT processing, IBM File Export for z/OS supports the following limited subset of SELECT. The basic SELECT syntax that is used in IBM File Export for z/OS is:

```
SELECT column_list FROM OBJECT WHERE(where predicate) WHEN(when predicate) PRINT
```

In this form of the SELECT, the items in the select list are placed in the output record in the order that they are named--that is, the first named column is placed in position one of the record and the second named column goes into the next acceptable byte. *Acceptable* means that the SYNC alignment is met according to the description of the source field. For example, if the second field is an integer, this second field is placed in the next byte of the output record that has integer alignment.

An alternative format for the select statement is as follows:

```
SELECT column_list INTO ofield_list FROM OBJECT WHERE(where predicate) WHEN(when predicate) PRINT
```

The *ofield_list* is restricted to a list of output fields (see "OFIELD" on page 225). In this form of SELECT, the first item in the select list is moved to the position described by the first field in the ofield_list, the second to the second, and so forth. The next source column is NOT moved to the next acceptable byte. Rather, it moves it to the position "pointed to" by the paired ofield list item.

Note: If a record that satisfies one or two selection criteria exists in the input stream, then the record is selected only once and no more matching is performed.

WHERE

The WHERE predicate can include the following elements:

```
UNIQUE ON (field-name) PRINT FILLER(BLANK|BLANKS,ZERO|ZEROES|x'00'...x'FF') FINAL  
MINIMUM(nn) MAXIMUM(nn) SPUFI(nnnnnnnn|*) QUALIFIER(owner_name) TABLE(table_name)  
NO_FORCE NO_EDIT INLIST;
```

UNIQUE ON

Use UNIQUE ON to specify that all of the records that are selected by this SELECT statement are unique. The syntax is as follows:

```
UNIQUE ON [field-name list]
```

Uniqueness is determined by evaluating whether field1 concatenated to field2 (and so on with all fields in the list) always produces a unique record. This is useful when it is necessary to reduce redundancies in the output that are produced by this SELECT statement.

PRINT

You can use PRINT to print the output returned by a SELECT statement. The syntax is:

```
PRINT(nn)
```

where *nn* is the number of records that you want to print. The default value for *nn* is 1. For example, PRINT(12) means print the first 12 records that are produced by the SELECT statement. All PRINT statements that are declared at the SEGMENT level are inherited at the TYPE level and subsequently at the SELECT level. If PRINT is declared at the TYPE level, it is inherited at the SELECT level. This element is optional. See also "Specifying MINIMUM, MAXIMUM, PRINT, or SPUFI" on page 262.

FILLER

```
FILLER(BLANK|BLANKS,ZERO|ZEROES|00...FF)
```

Use the FILLER declarative to specify what should be used as a slack byte filler or pad character. Use *BLANK* or *BLANKS* to specify that a blank should be used as the pad character. Use *ZERO* or *ZEROES* to specify that '0' should be used as the pad character. Use one of the 1-byte hexadecimal values 00 through FF to indicate that the specified hexadecimal value should be used as the pad character. If this is not specified, the *BLANK* or the filler as set in *OPTIONS* is used. This element is optional.

MINIMUM

Use the MINIMUM declarative to specify the minimum length of the record that is produced by the SELECT statement. The syntax is as follows:

```
MINIMUM(nn)
```

where *nn* is the minimum length of the record that is produced by the SELECT statement. Values that are shorter than the length specified will be padded out to the specified length using the value specified in the FILLER. All SELECT statements under a TYPE inherit the MINIMUM specified at the TYPE level. Specifying the MINIMUM at the SELECT level overrides the inherited value. A declaration of MINIMUM(0) negates the inherited value. If MINIMUM is declared at the SEGMENT level, it is inherited at the TYPE level and subsequently at the SELECT level. If MINIMUM is declared at the TYPE level, it is inherited at the SELECT level. In IMS mode it is recommended not to specify MINIMUM because the SEGMENT MINIMUM and MAXIMUM are taken from the definition found in the target (to) DBD and propagate to the TYPES and the SELECTS. However, you could specify this at the TYPE level (or SELECT level) if there is a required difference in the limits for a

particular type of record. This element is optional. See also “Specifying MINIMUM, MAXIMUM, PRINT, or SPUFI” on page 262.

MAXIMUM

Use the MAXIMUM declarative to specify the maximum length of the record that is produced by the SELECT statement. The syntax is as follows:

MAXIMUM(*nn*)

where *nn* is the maximum length of the record that is produced by the SELECT statement. Records that are produced by the SELECT statement that are longer than the specified length, result in an error at runtime. All SELECT statements under a TYPE inherit the MAXIMUM specified at the TYPE level. Specifying the MAXIMUM at the SELECT level overrides the inherited value. A declaration of MAXIMUM(0) negates the inherited value and suppresses the edits or printing. If MAXIMUM is declared at the SEGMENT level, it is inherited at the TYPE level and subsequently at the SELECT level. If MAXIMUM is declared at the TYPE level, it is inherited at the SELECT level. In IMS mode it is recommended not to specify MAXIMUM because the SEGMENT MINIMUM and MAXIMUM are taken from the definition found in the target (to) DBD and propagate to the TYPES and the SELECTS. However, you could specify this at the TYPE level (or SELECT level) if there is a required difference in the limits for a particular type of record. See also “Specifying MINIMUM, MAXIMUM, PRINT, or SPUFI” on page 262.

Information: UNIQUE ON (field-list), FILLER, PRINT, MINIMUM, and MAXIMUM are all optional and can be specified in any order, but must be declared after any predicates.

FINAL

Use the FINAL declarative to signify that this particular SELECT statement will decide that a segment set (from the preceding root to the next root) will be retained or discarded depending on whether or not it meets the predicates of this SELECT statement. This is valid when the OPTIONS(GROUP) declarative is used. This element is optional.

SPUFI

Use the SPUFI declarative to indicate that SPUFI formatted records (to the specified count) are to be produced. The syntax is as follows:

SPUFI(*nnnnnnnn* | *)

where *nnnnnnnn* represents the maximum number of records to be produced. The maximum value that can be specified is 2147483647. If asterisk (*) is specified, it is the same as specifying the maximum number, 2147483647. All SPUFI keywords declared at the SEGMENT level are inherited at the TYPE level and subsequently at the SELECT level. If a SPUFI keyword is declared at the TYPE level, it is inherited at the SELECT level. This element is optional, but if used, you must specify ABXSQL DD. See also “Specifying MINIMUM, MAXIMUM, PRINT, or SPUFI” on page 262.

QUALIFIER

You can use QUALIFIER to set the table qualifier and owner for any SPUFI output from the SELECT statement. The syntax is as follows:

QUALIFIER(*owner_name*)

where *owner_name* is the name of the table owner. All QUALIFIER statements declared at the SEGMENT level are inherited at the TYPE level and subsequently at the SELECT level. If QUALIFIER is declared at the TYPE level, it is inherited at the SELECT level. This element is optional.

TABLE

Use the TABLE declarative to set the table name for any SPUFI output. The syntax is as follows:

TABLE(*table_name*)

If the table name is not set, the object from which you are selecting in the SELECT statement (SELECT * FROM *object name*) when producing SPUFI output. This element is optional.

Information: All predicates (if any) must precede the PRINT, UNIQUE, QUALIFIER, SPUFI, MINIMUM, MAXIMUM, and TABLE QUALIFIER declaratives.

NO_FORCE

Use the NO_FORCE or NOFORCE declarative to prevent sequence field content from being forced to be the same as the matched sequence fields for the input segment and output segment when working with DLI data. This element is optional. By default, whenever there has been a change in a matched sequence field, the resultant record's sequence fields will be overlaid from the source image assuring that the content is the same (a message is issued whenever this occurs). To override this behavior, you can use the NOFORCE declarative. When you specify NOFORCE or NO_FORCE, editing becomes active. Editing compares the matched fields and terminates the run if there is a failure in the comparison.

NO_EDIT

Use the NO_EDIT or NOEDIT declarative to prevent matched sequence fields from being forced or edited. (Edit is activated when NOFORCE is specified when working with DLI data.) This element is optional.

Logical operators

Logical operations can be achieved using one of two logical operators. The following lists logical operations and the logical operators that can be used in the IBM File Export for z/OS command syntax to represent them:

Equal to

- EQ
- =

Less than or equal to

- LE
- <=

Less than

- LT
- <

Not equal to

- NE
- <>

Greater than

- GT
- >

Greater than or equal to

- GE
- >=

IN | INLIST

Use the IN or INLIST declarative to specify a criterion to be met in order for data to be selected. The syntax is as follows:

- IN (*literal*, *literal*, ...)
- IN_LIST
- IN (*source-name*) where the source is defined using a SOURCEDD or RELATIONSHIP statement.

You can also build a "not-in-list." For example, when working with DB2 data:

```
SELECT X FROM SOURCE WHERE PREDICATES = #1
SELECT NOTHING FROM SOURCE = #2
THEN INSERT INTO DD:NOTLIST FIELDX = #3
```

The THEN INSERT (#3) is executed whenever the first select (#1) fails (because SELECT NOTHING (#2) is always TRUE as it has no predicates). When the source is DB2, you build a 'NOT-IN-LIST' by reversing the predicates. You use WHERE NOT (predicates) and use the NOTHING expression to suppress writing the produced record. where *short-name* is max of 32 chars with table names up to 128 bytes.

SCRAMBLE

Use the SCRAMBLE declarative to specify that the data in a field should be randomized in the target. When SCRAMBLE is specified, the source data is replaced in the target by data that is different but matches the formatting of the source data. This is useful for protecting confidential data. The SCRAMBLE declarative is used within the FIELD declarative.

LITERAL

See "LITERAL" on page 222.

VARIABLE

See "VARIABLE" on page 254.

Information: You can specify a one byte field *and* a literal (or variable) as the predicate. The field can only be one byte and *all* bits must match (mixed bits are not supported). For example:

```
[Field_name] AND [literal]
```

or

```
[Field_name] AND [literal_name]
```

When you want to select no records:

You can use an additional type of SELECT statement:

```
SELECT NOTHING FROM object WHERE [where predicate]
```

This statement specifies that nothing from this record type or segment will be included in the target. It excludes the currently read segment, type, or record. This syntax can also be used to exclude the specified record, then insert a substitute by following it with: THEN INSERT [*insert predicate*] This allows you to add a record or segment at the place where this statement evaluates as true, while excluding the record that was just read. The excluded record can still be used as a COLLECT source, and fields in the record can be used in the following THEN INSERT statements, but this record is not written to the target output. These predicates can be useful in eliminating records from a file that are known to be faulty, if you know something about its content that is unique and can be expressed in predicates.

Note: To copy a whole record and scramble selected fields you can use the following syntax:

```
SELECT * SCRAMBLING(field1,field2, additional fields) FROM table name WHERE where clause
```

Using the Asterisk (*)

The asterisk has the following definition for variable input record:

- DLI variable input: from the data start (offset 2) through the end of the record. It requires a half word alignment in the output record.
- SEQUENTIAL variable input: from the start of data (offset 4) through the end of the record. It requires a word alignment in the output record.
- In all other cases the asterisk is the length of the input record and requires a double word alignment in the output record.

When going from fixed to variable or from variable to fixed, alignment issues can occur. In the case of DLI segments, any field that needs an alignment that is more restrictive than a halfword, will probably lose alignment. For a sequential variable this is true for any field requiring double word alignment.

SKIP

Use the SKIP declarative to specify that you want to skip the first *nn* records of the input file when SKIP is used globally, or to specify that *nn* records of the selected records for output should be skipped before output is written to the target. The syntax is as follows:

```
SKIP(nn)
```

where *nn* is the number of records that you want to skip. For IMS, this is per child within each unique parent. For sequential or VSAM records, this causes *nn* records in the input to be passed over, either at the TYPE level, or globally if SKIP is specified prior to any TYPE declaratives being specified.

See also “Limiting output record counts using SKIP and LIMIT” on page 258.

SOURCEDB2

Use this declarative to specify the name of the source DB2 database. This declarative is used only when working with DB2 data. The syntax is as follows:

```
SOURCEDB2(db2-subsystem-name);
```

where *db2-subsystem-name* is the name of the DB2 subsystem where the source data resides.

SOURCEDBD

Use the SOURCEDBD declarative to specify the name of a member that describes the input DBD format. The syntax is as follows:

```
SOURCEDBD(dbdname) GROUP;
```

dbdname

Name of a member that describes the input DBD format.

Depending on the type of IMS processing that is occurring, information about the source DBD is found as follows:

- When in IMS mode, IBM File Export looks in the DBDLIB pointed to by DD OLDIMS.
- When in DEDB mode, IBM File Export looks in the file pointed to by DD UR7DBDFN.

- When in DLI mode, IBM File Export looks in the DBDLIB pointed to by DD IMS.
- When in DBB mode, IBM File Export looks in the ACBLIB pointed to by IMSACB DD. MODSTAT is also supported.
- When in BMP mode, IBM File Export looks in the ACBLIB pointed to by IMSACB DD. MODSTAT is also supported.

GROUP

An optional keyword to indicate that selection for a set of segments is based on that set of segments that satisfies the FINAL SELECT. For additional information about the FINAL declarative, see FINAL 241.

GROUP may still be specified on the OPTIONS declarative. In that case, when processing multiple databases, it will appear as if GROUP were specified for each SOURCEDBD statement.

Note that FINAL must be specified in one of the SELECT statements that is subordinate to this SOURCEDBD before the next SOURCEDBD statement or the end-of-file is encountered.

SOURCEDD

The SOURCEDD declarative is no longer used but is retained for compatibility. It has been replaced by the RELATIONSHIP declarative.

SSM

Use the SSM declarative to specify the external subsystem identifier. This keyword is valid when DLI, DBB, or BMP is specified on the OPTIONS keyword. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for a list of restrictions and considerations regarding the use of the SSM keyword. The syntax is as follows:

```
SSM(ssmid);
```

where:

ssmid

is a 1- to 4- byte character name for the external subsystem identifier.

If the SSM keyword is specified in the control statement stream, it will override the SSM specification in the ABXOIMS0 option control block for this execution.

STIMER

Use the STIMER declarative to specify whether processor time statistics should be gathered. This keyword is valid when BMP is specified on the OPTIONS keyword. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for a list of restrictions and considerations regarding the use of the STIMER keyword. The syntax is as follows:

```
STIMER(0|1|2);
```

where:

0 indicates that no processor time statistics are to be gathered.

1 or 2 both indicate that processor time statistics are to be gathered, but different statistics may be accumulated based on other parameters. Refer to *IMS Installation Volume 2: System Definition and Tailoring* for these details.

If this keyword is specified in the control statement stream, it will override the STIMER specification in the ABXOIMS0 option control block for this execution.

SWAP

Use the SWAP declarative to specify whether the address space is swappable. This keyword is valid when DLI or DBB is specified on the OPTIONS keyword. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for a list of restrictions and considerations regarding the use of the IMSID keyword. The syntax is as follows:

```
SWAP(N|Y);
```

where:

N indicates that the address space is nonswappable.

Y indicates that the address space is swappable.

If this keyword is specified in the control statement stream, it will override the SWAP specification in the ABXOIMS0 option control block for this execution.

TABLE

Use the TABLE declarative, when working with DB2 data, to identify the source table (or tables) and the target table to be loaded into in the target DB2 system. You can also use the TABLE declarative to qualify which rows are selected through matching to relationship identities and to limit rows based on a parent-child relationship.

```
TABLE (short-name) SOURCE(owner.table, owner.table2, etc...) TARGET (owner.table)  
dd:ddname SORT (column1 asc|desc,column2 asc|desc) CHILD OF PARENT=short-name-of-  
parent WHERE THIS short-name-of-current-table.column1, column2 = short-name-of-  
parent.column1, column2 ONLY WHEN (column1) IN|NOT IN (source-name) AND|OR column2  
IN|NOT IN (source-name2) ACCEPT NULLS NOTHING LOAD(resume|replace|reuse) PRINT(nn);
```

where ONLY WHEN column IN can be replaced with ONLY WHEN column NOT IN (*source-name*) and *short-name* is the table name.

SOURCE:

Use the SOURCE declarative to define the source table or tables that you want to use. You can declare as many source tables as necessary using the SOURCE declarative. The syntax is as follows:

```
SOURCE(owner.table, owner.table, ...)
```

TARGET:

Use the TARGET declarative to define the TARGET table that you want to use. The syntax is as follows:

```
TARGET (owner.table)
```

dd:

Use the DD declarative to specify the ddname. The syntax is as follows:

```
DD:ddname
```

SORT:

Use the SORT declarative to specify the column that is to be sorted, and whether it is to be sorted in ascending (asc) or descending (desc) order. The syntax is as follows:

```
SORT (column1 asc|desc,column2 asc|desc)
```

Parent - child relationships:

You can define parent-child relationships within the TABLE declarative. The syntax is as follows:

```
CHILD OF PARENT= short-name WHERE THIS short-name.column1, col2 = PARENT
```

Only the first column in the parent child relationship is qualified-- in the parent field and the child field.

Performance note: The parent-child relationships live in memory. As a parent record is read and then selected, its key is preserved in memory until the associated child has been processed. If the keys are long, or if there are many parent records, this may take a large amount of memory. A method to abate this is in the use predicates at the parent and child level SELECT statements. In this case, no memory is needed.

For example: Let us say that you wanted content from SYSIBM.SYSTABLES and SYSIBM.SYSCOLUMNS and the child-parent relationship is on `child.tbcreator, tbname = parent.creator, name`. Then the following would select the associated rows in concert, without using the application-defined relationships. At the parent:

```
SELECT * from SYSIBM.SYSTABLES where creator <> 'SYSIBM';
```

Then at the child:

```
SELECT * from SYSIBM.SYSCOLUMNS where tbcreator <> 'SYSIBM';
```

ACCEPT NULLS:

Use the ACCEPT NULLS declarative to specify that a column should be accepted if it has a null value. By definition, a NULL column has no value and can only be evaluated with 'column is null' or 'column is not null'. By default, a fetched row with the pertinent column having a null value can not be evaluated as being in or not in a list of values and as such is passed over. Specifying ACCEPT NULLS causes the record to be accepted if the column is null (that is IN or NOT IN are evaluated as TRUE if the content is null). If ACCEPT NULLS is not specified, the record is passed over.

NOTHING:

Use the NOTHING expression to suppress writing the record that is produced by the SELECT statement.

LOAD:

Use the LOAD declarative to specify the load setting for this table. The syntax is as follows:

```
LOAD(resume|replace|reuse)
```

You can specify:

- `resume` indicates whether records are to be loaded into an empty or non-empty table space.
- `replace` indicates whether the table space and its indexes need to be reset to empty before records are loaded.
- `reuse` is used with `replace` and specifies that LOAD should logically reset and reuse DB2-managed data sets without deleting and redefining them.

This declarative is optional.

TARGETDB2

Use this declarative to specify the name of the target DB2 database. The syntax is as follows:

```
TARGETDB2(db2-subsystem-name);
```

where *db2-subsystem-name* is the name of the DB2 subsystem to which the target data will be written.

TARGETDBD

Use the TARGETDBD declarative to specify the name of a member that describes the output DBD format. If this value is not provided, the SOURCEDBD name is used. The syntax is as follows:

```
TARGETDBD(dbdname);
```

dbdname

Name of a member that describes the output DBD format.

Depending on the type of IMS processing that is occurring, information about the source DBD is found as follows:

- When in IMS mode, IBM File Export looks in the DBDLIB pointed to by DD NEWIMS.
- When in DLI mode, IBM File Export looks in the DBDLIB pointed to by DD IMS.
- When in DEDB mode, IBM File Export looks in the ACBLIB pointed to by DD ACBLIB. MODSTAT is also supported.
- When in DBB mode, IBM File Export looks in the ACBLIB pointed to by DD IMSACB. MODSTAT is also supported.
- When in BMP mode, IBM File Export looks in the ACBLIB pointed to by DD IMSACB. MODSTAT is also supported.

THEN INSERT

Use THEN INSERT when you want to extract records from one database or file and then extract records from another database or file that match this first set of records. As an example, you may want to build a test database that has matched records from payroll and personnel sources based on a 'key' such as employee identifier (this need not be an actual key). To do so, it is necessary to keep track of the information from the first extract, identifying what was selected or extracted, and then to drive a subsequent extract based on the record of the first extraction. THEN INSERT INTO DD:*ddname* is used to keep track of what the first extract selected.

The basic THEN INSERT syntax that is used in IBM File Export for z/OS is:

```
THEN INSERT INTO object VALUES(value name, value name 2, value name x) PRINT(nn)  
FILLER(BLANK|BLANKS,ZERO|ZEROES|x'00'...x'FF') MINIMUM(nn) MAXIMUM(nnn)  
QUALIFIER(owner) TABLE(table name) JCL VSAM
```

where PRINT, FILLER, MINIMUM, QUALIFIER, TABLE, JCL, and VSAM are optional.

THEN INSERT is not valid in LOAD mode. It is typically anchored off of a SELECT statement and is executed only when the preceding SELECT is true. THEN INSERT supports the insertion of literals, named literals, variables, fields, and collects. THEN INSERT can take the following forms:

1. THEN INSERT INTO DD:*ddname*...;
2. THEN INSERT INTO ABXOUT...;
3. THEN INSERT *ofield1*, *ofield2* INTO DD:*ddname* VALUES(*value list*)...;
4. THEN INSERT *ofield1*, *ofield2* INTO ABXOUT VALUES(*value list*)...;

Use the INSERT INTO DD:*ddname* declarative to specify that data should be inserted into the specified target rather than into ABXOUT. *ddname* is the name of the DD where you want the data to be inserted. This THEN INSERT statement does not place data into ABXOUT, but rather into the named target: *ddname*. (When you create the *ddname* data set, the DCB parameters in the JCL should be omitted. The RECFM will be set to FB and the LRECL and BLKSIZE will be set as required by the data values that are produced by the INSERT statement.) You can have multiple INSERT INTO DD:*ddname* statements, but those that point to the same *ddname* must have the same LRECL and be the same logical data.

In formats 1 and 2, above, what is inserted is a column list that may contain field names, collects, literals, and so forth. In this instance, the first item in the column list is moved to position one of the record, the second item into the next acceptable byte, and so forth. *Acceptable* means that the SYNC alignment is met according to the description of the source field.

In formats 3 and 4, above, what is inserted is an *ofield_list*. Such a list is restricted to a list only of output fields (see "OFIELD" on page 225). In this instance, the first item in the column_list is moved to the position described by the first field in the *ofield_list*, the second to the second, and so forth. The next source column is NOT moved to the next acceptable byte. Rather, it moves it to the position "pointed to" by the paired *ofield_list* item. Using this format for the THEN INSERT statement provides you with better control over the produced record's format.

The THEN INSERT predicate can include the following elements:

VALUES

The value-list can contain:

- columns (if you are using a DB2 source)
- fields (if you are using a non-DB2 source)
- named literals
- literals
- variables
- collects (if you are using a non-DB2 source).

PRINT

You can use PRINT to print the contents produced by a INSERT statement.

The syntax is:

```
PRINT (nn)
```

where the default is for *nn* is 1. For example, PRINT(12) means print the first 12 records that are inserted by the INSERT statement. All PRINT statements that are declared at the SEGMENT level are inherited at the TYPE level and subsequently at the INSERT level. If PRINT is declared at the TYPE level, it is inherited at the INSERT level. This element is optional. See also "Specifying MINIMUM, MAXIMUM, PRINT, or SPUIFI" on page 262.

FILLER

```
FILLER(BLANK|BLANKS,ZERO|ZEROES|00...FF)
```

Use the FILLER declarative to specify what should be used as a slack byte filler or pad character. Use *BLANK* or *BLANKS* to specify that a blank should be used as the pad character. Use *ZERO* or *ZEROES* to specify that '0' should

be used as the pad character. Use one of the 1-byte hexadecimal values 00 through FF to indicate that the specified hexadecimal value should be used as the pad character. If this is not specified, the BLANK or the filler as set in OPTIONS is used. This element is optional.

MINIMUM

You can use MINIMUM to specify the minimum length of the record that is produced by the SELECT statement. The syntax is as follows:

MINIMUM(*nn*)

Values that are shorter than the length specified will be padded out to the specified length using the filler you have specified in the FILLER element. All INSERT statements under a TYPE inherit the MINIMUM specified at the TYPE level. Specifying the MINIMUM at the INSERT level overrides the inherited value. A declaration of MINIMUM(0) negates the inherited value and suppresses the edits or printing. If MINIMUM is declared at the SEGMENT level, it is inherited at the TYPE level and subsequently at the INSERT level. If MINIMUM is declared at the TYPE level, it is inherited at the INSERT level. In IMS mode, it is recommended not to specify MINIMUM because the SEGMENT MINIMUM and MAXIMUM are taken from the definition found in the target (to) DBD and propagate to the TYPES and the INSERTS. However, you could specify this at the TYPE level (or INSERT level) if there is a required difference in the limits for a particular type of record. This element is optional. See also “Specifying MINIMUM, MAXIMUM, PRINT, or SPUIFI” on page 262.

MAXIMUM

You can use MAXIMUM to specify the maximum length of the record that is produced by the SELECT statement. The syntax is as follows:

MAXIMUM(*nn*)

Values that are longer than the specified length result in an error at runtime. All INSERT statements under a TYPE inherit the MAXIMUM specified at the TYPE level. Specifying the MAXIMUM at the INSERT level overrides the inherited value. A declaration of MAXIMUM(0) negates the inherited value and suppresses the edits or printing. If MAXIMUM is declared at the SEGMENT level, it is inherited at the TYPE level and subsequently at the INSERT level. If MAXIMUM is declared at the TYPE level, it is inherited at the INSERT level. In IMS mode it is recommended not to specify MAXIMUM because the SEGMENT MINIMUM and MAXIMUM are taken from the definition found in the target (to) DBD and propagate to the TYPES and the INSERTS. However, you could specify MAXIMUM at the TYPE level (or INSERT level) if there is a required difference in the limits for a particular type of record. See also “Specifying MINIMUM, MAXIMUM, PRINT, or SPUIFI” on page 262.

Information: UNIQUE ON (field-list), FILLER, PRINT(*nn*), MINIMUM(*nn*), and MAXIMUM(*nn*) are all optional and can be specified in any order, but must be declared after any predicates.

SPUIFI

Use the SPUIFI declarative to indicate that SPUIFI formatted records (to the specified count) are to be produced. The syntax is as follows:

SPUIFI (*nnnnnnnnn* | *)

where *nnnnnnnnnn* represents the maximum number of records to be produced. The maximum value that can be specified is 2147483647. If asterisk (*) is specified, it is the same as specifying the maximum number, 2147483647. All SPUIFI keywords declared at the SEGMENT level are inherited at the TYPE level and subsequently at the SELECT level. If a SPUIFI keyword is declared at the TYPE level, it is inherited at the SELECT level. This

element is optional, but if used, you must specify ABXSQL DD. See also “Specifying MINIMUM, MAXIMUM, PRINT, or SPUFI” on page 262.

QUALIFIER

Use the QUALIFIER declarative to set the table qualifier and owner for any SPUFI output from the INSERT statement. The syntax is as follows:

```
QUALIFIER(owner_name)
```

All QUALIFIER statements declared at the SEGMENT level are inherited at the TYPE level and subsequently at the INSERT level. If QUALIFIER is declared at the TYPE level, it is inherited at the INSERT level. This element is optional.

TABLE

TABLE is valid for DB2 only. Use the TABLE declarative to identify the source table (or tables) and the target table to be loaded into in the target DB2 system. The syntax is as follows:

```
TABLE(table_name)
```

where *table_name* is the name that identifies the table. A table named at the SEGMENT level is inherited at the TYPE level and subsequently at the SELECT level. If a table is named at the TYPE level, it is inherited at the SELECT level.

Requirements associated with the THEN INSERT statement in IBM File Export for z/OS

The following requirements must be taken into consideration when creating THEN INSERT statements for use in IBM File Export for z/OS:

- When working with IMS databases, objects must be segments in the TARGETDBD.
- When using RRDS as the target of

```
THEN INSERT INTO DD:DDNAME VALUES (A,B,any other values)...
```

the first field, 'A' in this example, must be a four byte binary zero (or an integer of zero value). The second field must be an integer with the record number (which must be greater than zero, usually commencing with 1). This can be achieved as follows:

```
Variable(recnumber) = 1 (4,int);
```

```
THEN INSERT INTO DD:DDNAME VALUES( x'00000000' , recnumber , any other values)
```

- If you have multiple INSERT INTO DD:*ddname* statements that point to the same *ddname*, you must specify the same LRECL and use the same logical data.

TYPE

When you are working with sequential, VSAM, or DL/I data, a file may have different record types interspersed in the data. With DL/I data, this can apply at the SEGMENT level. Use the TYPE declarative to specify the type of records that require special or alternative processing. (Record types are similar to COBOL redefine definitions.) You identify the type of records with the TYPE statement which formulates a predicate or predicate group to qualify the record type. Associated with this typing, specific selects and limits and skipping and THEN INSERTs may be created. In essence, you can specify specific processing options for record types such as debit records versus credit records.

You can specify MINIMUM, MAXIMUM, PRINT, SPUFI, and QUALIFIER within the TYPE declarative. These must be specified after any predicates. The TYPE name must be unique. Names (fields) used in predicates must be defined at the SEGMENT or

Global level. If you are working with IMS data and you are using SEGMENTS, the TYPE declarative must appear after the SEGMENT to which it applies. The syntax is as follows:

```
TYPE(type_name WHEN(when_predicate AND|OR when_predicate)|WHERE (where_predicate)
MINIMUM(nn) MAXIMUM(nn) PRINT(nn) TABLE(table_name) SPUFI(nnnnnnnn|*)
QUALIFIER(owner_name))
```

where *when_predicate* is:

```
WHEN name operator name
```

where *operator* can be:

- LIKE
- INLIST
- A logical operator:
 - EQ
 - =
 - LE
 - <=
 - LT
 - <
 - NE
 - <>
 - GT
 - >
 - GE
 - >=

The first predicate that evaluates as true sets the TYPE. Therefore, the most restrictive TYPE should be first. There is a special TYPE named DEFAULT whose syntax is TYPE(DEFAULT). It must be the last TYPE definition in the list -- within the definitions or within the SEGMENT. This type name can be used more than once. At runtime, if no type is matched, the TYPE is set to: 'SELECT *'. To override this behavior, use the following statements following all of the other TYPE declaratives:

```
TYPE(DEFAULT);
```

```
EXCLUDE;
```

If you have a SELECT statement preceding a first explicit TYPE declaration within a segment, TYPE(DEFAULT) is created. This precludes an explicit type of DEFAULT. In general, an initial SELECT followed by an explicit type would only be used when you want no default type on the initial SELECT predicate. You can specify MINIMUM, MAXIMUM, PRINT, SPUFI, and QUALIFIER within the TYPE declarative.

MINIMUM

Use the MINIMUM declarative to specify the minimum length of the record that is produced by the SELECT statement from this SEGMENT. The syntax is as follows:

```
MINIMUM(nn)
```

where *nn* is the minimum length of the record that is produced for records of this TYPE. Values that are shorter than the length specified will be padded out to the specified length using the value specified in the FILLER. All SELECT statements under a TYPE inherit the MINIMUM specified at the

TYPE level. Specifying the MINIMUM at the SELECT level overrides the inherited value. A declaration of MINIMUM(0) negates the inherited value. If MINIMUM is declared at the SEGMENT level, it is inherited at the TYPE level and subsequently at the SELECT level. If MINIMUM is declared at the TYPE level, it is inherited at the SELECT level. In IMS mode, it is recommended that MINIMUM not be specified because the SEGMENT MINIMUM and MAXIMUM are taken from the definition found in the target (to) DBD and propagate to the TYPES and the SELECTS. However, you could specify this at the TYPE level (or SELECT level) if there is a required difference in the limits for a particular type of record. This element is optional. See also “Specifying MINIMUM, MAXIMUM, PRINT, or SPUFI” on page 262.

MAXIMUM

Use the MAXIMUM declarative to specify the maximum length of the record that is produced by the SELECT statement from this SEGMENT. The syntax is as follows:

```
MAXIMUM(nn)
```

where *nn* is the maximum length of the record that is produced for records of this TYPE. Records that are produced by the SELECT statement that are longer than the specified length, result in an error at runtime. All SELECT statements under a TYPE inherit the MAXIMUM specified at the TYPE level. Specifying the MAXIMUM at the SELECT level overrides the inherited value. A declaration of MAXIMUM(0) negates the inherited value and suppresses the edits or printing. If MAXIMUM is declared at the SEGMENT level, it is inherited at the TYPE level and subsequently at the SELECT level. If MAXIMUM is declared at the TYPE level, it is inherited at the SELECT level. In IMS mode, it is recommended that MAXIMUM not be specified because the SEGMENT MINIMUM and MAXIMUM are taken from the definition found in the target (to) DBD and propagate to the TYPES and the SELECTS. However, you could specify this at the TYPE level (or SELECT level) if there is a required difference in the limits for a particular type of record. See also “Specifying MINIMUM, MAXIMUM, PRINT, or SPUFI” on page 262.

PRINT

You can use PRINT to print the output returned for records of this TYPE. The syntax is:

```
PRINT(nn)
```

where *nn* is the number of records that you want to print. The default value for *nn* is 1. For example, PRINT(12) means print the first 12 records that are produced by the SELECT statement from this SEGMENT. All PRINT statements that are declared at the SEGMENT level are inherited at the TYPE level and subsequently at the SELECT level. If PRINT is declared at the TYPE level, it is inherited at the SELECT level. This element is optional. See also “Specifying MINIMUM, MAXIMUM, PRINT, or SPUFI” on page 262.

TABLE

TABLE is valid for DB2 only. Use the TABLE declarative to identify the source table (or tables) and the target table to be loaded into in the target DB2 system. The syntax is as follows:

```
TABLE(table_name)
```

where *table_name* is the name that identifies the table. A table named at the SEGMENT level is inherited at the TYPE level and subsequently at the SELECT level. If a table is named at the TYPE level, it is inherited at the SELECT level.

SPUFI

Use the SPUFI declarative to indicate that SPUFI formatted records (to the specified count) are to be produced. The syntax is as follows:

```
SPUFI(nnnnnnnn |*)
```

where *nnnnnnnnnn* represents the maximum number of records to be produced. The maximum value that can be specified is 2147483647. If asterisk (*) is specified, it is the same as specifying the maximum number, 2147483647. All SPUFI keywords declared at the SEGMENT level are inherited at the TYPE level and subsequently at the SELECT level. If a SPUFI keyword is declared at the TYPE level, it is inherited at the SELECT level. This element is optional, but if used, you must specify ABXSQL DD. See also “Specifying MINIMUM, MAXIMUM, PRINT, or SPUFI” on page 262.

QUALIFIER

You can use QUALIFIER to set the table qualifier and owner for any SPUFI output for records of this TYPE. The syntax is as follows:

```
QUALIFIER(owner_name)
```

where *owner_name* is the name of the table owner. All QUALIFIER statements declared at the SEGMENT level are inherited at the TYPE level and subsequently at the SELECT level. If QUALIFIER is declared at the TYPE level, it is inherited at the SELECT level. This element is optional.

VARIABLE

A VARIABLE is a LITERAL that changes value (is incremented) by the system at points in the processing. As an example, it could be a unique ordinal for each output record which gives that record a unique identity. These are particularly useful in LOAD processing to create test data that has unique content by record. Use the VARIABLE declarative to create a literal that increments or increases in value. In implementation a VARIABLE is a LITERAL and can be used wherever the syntax allows a literal or named literal. The VARIABLE declarative must appear after the LITERAL declarative. The syntax is as follows:

```
VARIABLE (literal-name) = literal-value (length,data_type) BY nnn RESET USED
```

where: *length,data_type*, BY *nnn*, RESET, and USED are optional and:

literal_name

The name for the literal that you are defining. The name must be unique within the IBM File Export for z/OS control statement syntax.

literal-value

Definition

length

The length of the literal. It is a numeric integer type constant. The length of the bytes required in memory for the field. The length can also be specified as '*' (an asterisk) which means the rest of the record as seen at run time. The sum of offset and length cannot exceed the record length. This declarative is optional.

data_type

The data type of the field. (For a description of these data types, see Table 3 on page 257.) You can specify any of the following types:

- CHAR
- INT or INTEGER
- SHORT
- HEX
- PACKED
- FLOAT
- DOUBLE

This declarative is optional.

BY Use the BY declarative to specify the value by which the literal will be incremented. The syntax is as follows:

BY *nnn*

If this is not specified, the default value of 1 will be used. This declarative is only used when working with FLOAT, DOUBLE, or INTEGER data.

RESET

Use the RESET declarative to reset the value of the VARIABLE to its initial value on each root (in non-LOAD mode). This re-initializes the VARIABLE to its value as defined at the occurrence of a root. This is useful for values that are part of a family of values that increment. IBM File Export for z/OS keeps track of the use of variables in SELECT and INSERT statements and resets rescuable variables when navigating down to the first level of reference. If variables are used to populate sequence fields, you must use the NO_EDIT declarative in the SELECT statement. This declarative is only used when working with DLI data.

USED Use the USED declarative to increment a VARIABLE each time it is referenced. This is particularly valuable for putting some variable content in TWINS to make something unique in the record. However, if you use the USED declarative in this way, reserve this particular variable for this twin set as any 'use' will increment. This declarative is only used when working with DLI data.

When a VARIABLE is used, the value of the literal that is specified is incremented through a range as follows:

- An INTEGER value is incremented by 1 (or the values specified in the BY declarative) up to two billion. Once the value reaches two billion, it returns to zero and begins the cycle again.
- A SHORT value is incremented by 1 up to 32767. Once the value reaches 32767, it returns to zero and begins the cycle again.
- FLOAT and DOUBLE are incremented by 1 or the specified value.
- PACKED values are incremented by 1 until reaching 9 and then return to 1.
- HEX values are incremented by binary 1 to its limit of 'ff', then return to zero.
- CHAR values are incremented through the array of ' ', a to z, A to Z, and 0 to 9.

When a value is incremented, the right-most component is increased and carried to the left through the field. In order for intermeditation to occur, the following criteria must be met:

- The VARIABLE is used in a SELECT, INSERT, or THEN INSERT statement that results in a record being created
- The VARIABLE is referenced in a predicate that causes intermeditation

Incrementing occurs:

- In LOAD mode, after all the INSERTs have been executed once
- When working with VSAM or SEQUENTIAL data, on each record read
- When working with DLI data not in LOAD mode, on the occurrence of the second and subsequent roots.

WUNIT

Use the WUNIT declarative to specify the esoteric name of the devices that should be used to hold any new or temporary data sets.. This keyword is valid for all processing options. The syntax is as follows:

```
WUNIT(name);
```

where:

name is a 1- to 8- byte value that will be used as the parameter of the UNIT= keyword in JCL and dynamic allocation when allocating new or temporary data sets.

If the WUNIT keyword is specified in the control statement stream, it will override the WUNIT specification in the ABXOGBL0 option control block for this execution.

Chapter 11. Additional Information about Control Statement Syntax

This chapter contains additional information you will require when you choose to use IBM File Export's control statements directly rather than through the ISPF user interface.

Including comments in your JCL

Two types of comments are supported: printing and non-printing. To cause a comment to be printable on ABXPRINT, place a single colon (:) in column 1 of a comment statement. To cause a comment to be non-printable, place a double colon (::) in columns 1 and 2 of a comment statement.

About scrambling

When an object is scrambled, the contents will retain the same format as in the source, but the contents will be randomized. This is useful when the source contains sensitive data. The algorithm used for scrambling produces reproducible output. This means that the output of a scrambled value will be the same each time that value is scrambled.

Specification of IBM File Export data types

The table that follows provides information about the data types you specify either in the IBM File Export user interface or in the JCL you create.

Table 3. IBM File Export Data Types

Data Type	Memory Occupied	Alignment
INT or INTEGER	4 bytes	Fullword
CHAR	1 byte	Byte
FLOAT	4 bytes	Fullword
HEX	1 byte	Byte
SHORT	2 bytes	Halfword
DOUBLE	8 bytes	Doubleword
PACKED	1 byte	Byte

Understanding scope

Scope may be defined as being either global or local. An object that is of global scope affects all of the declaratives that follow it, while an object whose scope is local affects only the declarative on which it appears.

Scope of fields, literals, and variables

ABXUTIL uses the concept of scope to prevent names from colliding. A field defined on a SEGMENT statement has a scope of that segment (that is, until the next segment is defined). A field defined on a TYPE statement has the scope of that type.

So a name is first resolved within the TYPE statement, then within the SEGMENT statement, and finally from names defined in control statements appearing in the JCL prior to the first SEGMENT statement. Fields defined on a COLLECT statement are global in scope.

Scope of SKIP and LIMIT

SKIP and LIMIT control statements can have a global scope when specified for sequential or VSAM source files. However, there are some considerations when specifying SKIP or LIMIT as having global scope. These considerations are discussed in the next section.

Limiting output record counts using SKIP and LIMIT

SKIP and LIMIT control statements, when used globally, apply only to sequential or VSAM input (source) data. In all other cases, SKIP and LIMIT apply **to output candidate records only** and affect the data that is written to the target.

Global SKIP: SKIP, when used globally, indicates how many records are to be skipped when reading the input file.

Non-Global SKIP: When used other than globally, SKIP can appear on the TYPE and SELECT statements. On these statements, SKIP indicates how many records, out of all of the records selected for output, are to be skipped or omitted when writing data to the target.

Global LIMIT: LIMIT, when used globally, specifies the maximum number of input records to be selected.

Non-Global LIMIT: When used other than globally, LIMIT can appear on the TYPE and SELECT statements (VSAM and sequential) or the SEGMENT and TYPE levels (IMS), and specifies the maximum number of output records, out of all the records selected, that are to be written to the target. When the value specified on the LIMIT statement is reached, THEN ONE IN is implemented if it is specified (see "THEN ONE IN mmm" on page 259).

Global specification of SKIP and LIMIT

As noted above, SKIP and LIMIT control statements, when used globally, apply only to sequential or VSAM input (source) data. Note, however, that using SKIP and LIMIT globally does have some risks.

Assume, for example, that you have a sequential file containing 15,000 records, with mixed record types, from which you will extract 5000 records. The input file contains five types of records. For testing purposes, it is critical that your target data contain at least 200 of each type of record from the input data. You specify SKIP(1000) LIMIT(5000), causing you to skip the first 1000 input records and select a total of 5000 records. This may cause you to totally omit a specific record type if that record type appears only in the first 1000 records in the file, or conversely, appears only in the last 10,000 records.

In this case, you will want to specify SKIP and LIMIT at the TYPE level to ensure that you are affecting the types of records that are selected to be written to the target.

Non-global specification of SKIP and LIMIT

SKIP and LIMIT, when not used globally (that is, they do not appear in the JCL stream ahead of any SEGMENT, TYPE, and/or SELECT statement), apply as follows:

- For sequential and VSAM data, SKIP and LIMIT can be specified at the TYPE and SELECT levels.
For IMS data, SKIP and LIMIT can be specified only at the TYPE or SEGMENT level. In IMS mode, this value applies per child within each unique parent. The count is recycled each time the SEGMENT level is entered.
- For DB2 data, SKIP and LIMIT can be specified only at the TABLE level and filter the output data stream only after any CHILD OF or ONLY WHEN IN processing has created the output data stream.

THEN ONE IN *mmm*

If you want to continue to write selected records from the output to the target after you have reached the value you specified as LIMIT, you can further modify the LIMIT statement by adding THEN ONE IN *mmm*. The value you specify for *mmm* indicates that, after the LIMIT value is reached, write to the target every *mmm*th record. Thus, for example, if you specify THEN ONE IN 100, IBM File Export will write out the number of records that meets the LIMIT value, then continue to write out every 100th record thereafter.

Note that LIMIT 0 is not supported. However, you *can* specify LIMIT 0 THEN ONE IN *mmm* when you want to write out only every *mmm*th record.

Inserting data into IMS databases using the File Export UNLOAD file

IBM File Export for z/OS permits you to extract segments from multiple IMS databases and write the data into a sequential dataset, referred to here as the File Export UNLOAD file. Once you have produced the File Export UNLOAD file using ABXUTIL, you can insert the selected records into the appropriate IMS databases using program ABXFELD0.

About ABXFELD0

Using a control syntax that is similar to that of ABXUTIL, ABXFELD0 permits you to insert IMS segment data into IMS databases using the services of the DLI, DBB, or BMP processing mode. You can limit the databases that are to be processed by providing an INCLUDE and EXCLUDE mechanism. You can also read segment data from a database and write that data to a differently-named database.

Requirements and restrictions

- ABXFELD0 must be executed as an APF-Authorized program.
- ABXFELD0 must be supplied with an input file created by ABXUTIL when executed with an output mode of FEUNLOAD.
- ABXFELD0 does not support static PSBs; therefore, the Dynamic PSB feature must be available for use.

JCL requirements

The following JCL requirements must be satisfied in order to run ABXFELD0:

- ABXPRINT, ABXIN, and ABXCTL are the default names for the messages print output data set, the data input data set and the control statement input data set, respectively. You may need to customize your JCL to reflect any changes made to these ddnames by your enterprise during the IBM File Export installation and customization process.
- The IEFRDER DD is an output data set that contains the IMS database update logging data. This data set is required by IMS services when processing using DLI or DBB mode. It is not required if processing using BMP mode.

Example JCL

```
//STEP01 EXEC PGM=ABXFELD0,REGION=4M
//STEPLIBDDDSN=ABX.V110.SABXLOAD
//ABXPRINTDDSYSOUT=*
//ABXINDD DISP=SHR,DSN=abx.feunload.input.file
//IEFRDERDDDISP=(NEW,CATLG),DSN=ims.log.file
//ABXCTLD*
```

ABXFELD0 control statements

The FEUNLOAD file processor program ABXFELD0 uses control statement syntax similar to that of ABXUTIL. With the exception of the INCLDBD and EXCLDBD statements which are used exclusively by ABXFELD, and the OPTIONS statement which uses a subset of parameters available to program ABXUTIL, all control statements have the same requirements and restrictions as when used with program ABXUTIL. See Chapter 10, "Creating JCL Using Control Statements," on page 203 for guidance on their usage.

Valid control statement keywords are:

OPTIONS, IMSID, RESLIB1, RESLIB2, RESLIB3, DOPTLIB, DFSVSAMP, DPSBMAC, MON, DBDLIB, DBRC, SWAP, IRLMNAME, IRLM, BKO, DLTDUPS, AUDIT, NBA, OBA, CPUTIME, LOCKMAX, STIMER, PARDLI, DIRCA, DPSBPFY, DPSBSFX, BUF, EXCPVR, WUNIT, SSM, ALTID, FMTO, ALTRRCOO, GSGNAME, IMSPLEX, AGN, OPT, INCLDBD and EXCLDBD.

OPTIONS control statement

The available parameters for the OPTIONS control statement are as follows:

```
OPTIONS ( DLI|DBB|BMP);
```

No other parameters are permitted.

EXCLDBD control statement

The EXCLDBD control statement provides a method of preventing the insertion of data to the named database. Only one database per statement is permitted, but multiple occurrences of this keyword may be supplied. This keyword may only be supplied:

- If no INCLDBD control statement was supplied, or
- If the INCLDBD control statement had a parameter of ALL

The format of the EXCLDBD control statement is as follows:

```
EXCLDBD(dbd name);
```

INCLDBD control statement

The INCLDBD control statement provides a method of restricting the insertion of data to the named database. It also supports changing the database name when performing an insert. This to allow a single execution of ABXUTIL to produce an FEUNLOAD file that may be used to populate many differently-named databases that have the same database schema. The parameter ALL is also supported and must be the only parameter supplied if any EXCLDBD control statements are provided.

To process all databases using their original names, use the following format:

```
INCLDBD(ALL);
```

The following format will process database TEST01 only:

```
INCLDBD(TEST01);
```

The following format will process databases TEST01 and TEST02 only:

```
INCLDBD(TEST01);  
INCLDBD(TEST02);
```

The following format is used to read data for database TEST01 only and insert the data into database NEWDATA:

```
INCLDBD((TEST01),(NEWDATA));
```

Working with variable length records

The following sections outline some considerations when working with variable length records

Variable length input records and the use of the asterisk (*)

The asterisk has the following definition for variable length input records:

- DLI variable input: from the data start (offset 2) through the end of the record. It requires a half word alignment in the output record.
- SEQUENTIAL variable input: from the start of data (offset 4) through the end of the record. It requires a word alignment in the output record.
- In all other cases the asterisk is the length of the input record and requires a double word alignment in the output record.

When going from fixed to variable or from variable to fixed, alignment issues can occur. In the case of DLI segments, any field that needs an alignment that is more restrictive than a halfword, will probably lose alignment. For a sequential variable this is true for any field requiring double word alignment.

Using slack bytes when creating output files

When a record is inserted or formatted (by an INSERT, THEN INSERT, or SELECT statement), slack bytes as needed for alignment purposes are inserted preceding the asterisk for the following data types: INTEGER, SHORT, FLOAT, and DOUBLE. The default slack byte filler value is binary zero. The value you want to use for the slack byte can be set globally on the ABX\$GOPT panel. You can temporarily override this value by specifying a different value for FILLER on the INSERT or SELECT statement.

Understanding INSERT and THEN INSERT

INSERT is valid in LOAD mode only. There is no input file for an INSERT. INSERT supports the insertion of literals, named literals, and variables only. The INSERT statement takes the form `INSERT INTO insert_predicate; .`

THEN INSERT is not valid in LOAD mode. It is typically anchored off of a SELECT statement and is executed only when the preceding SELECT is true. THEN INSERT supports the insertion of literals, named literals, variables, fields, and collects. THEN INSERT can take the following forms:

- `THEN INSERT INTO DD:DDNAME...;`
- `THEN INSERT INTO ABXOUT...;`
- `THEN INSERT ofield1, ofield2 INTO DD:DDNAME VALUES(value list)...;`
- `THEN INSERT ofield1, ofield2 INTO ABXOUT VALUES(value list)...;`

Understanding Occurs Depending On

In treating data that is not of the second normal form, you may encounter repeating groups that occur several times in a single record. An example might be monthly data that occurs twelve times, one occurrence per month for a year. Sometimes the number of occurrences can vary. This is exemplified by the COBOL phrase `OCCURS 1 TO 20 TIMES DEPENDING ON something`. This *something* is called the object of the OCCURS DEPENDING ON or the ODO.

With the use of OCCURS DEPENDING ON, the position of fields to the right of the variable length repeating group in an expression (that is, the fields after the first instance of a variable length item) do not have a fixed position. Similarly, the length of a group item containing variable length data is not a constant. This type of data requires an algorithm to express the position of a field, and, in some cases, the length of a field or group item.

Specifying MINIMUM, MAXIMUM, PRINT, or SPUFI

When specifying MINIMUM, MAXIMUM, PRINT, or SPUFI at any other than the SELECT level, the value specified by *nnn* is inherited in any SELECT statements that follow. Thus, if you specify PRINT(10) for example, followed by 2 SELECTs, you will get up to 20 records; that is, up to 10 for each of the two SELECTs.

Chapter 12. Formatting Output Records

If you wish to format or reformat records being written to a target file or database, follow the guidelines presented here. Note that output written when using COBOL differs from the information provided below and is discussed in "How use of the Cobol parser affects output alignment" on page 264.

Output record formatting

Records are formatted from the SELECT list or from the INSERT VALUES list. Records are built in the order in which they appear in the list.

- Data that is defined as FLOAT, DOUBLE, INTEGER and SHORT is automatically aligned to its native boundaries.
- Data defined as STAR(*) is set on a double word boundary for Sequential, VSAM, and non-variable length DLI.
- Data defined as CHAR, HEX, or PACKED is not aligned.

There are three keywords that can adjust alignment beyond this native alignment: WORD, HWORD and DOUBLE. The use of these words in the SELECT or VALUES list causes the next field to align as specified. For example:

```
'SELECT "xxx" , field1 '
```

places field1 (CHAR or HEX or PACKED) in byte 4 of the record because "xxx" is three bytes long.

To place field 1 on a word boundary (byte 5), use either one of these two examples:

```
'SELECT "xxx",WORD,field1 '
```

```
'SELECT "xxx",HWORD,field1 '
```

When literals are used, they are taken at face value.

- A string literal has its assumed length; "HELLO" has a length of 5.
- Numeric literals have two formats: numeric and real. A literal that appears to be an INTEGER, 123 for example, is placed in an INTEGER. A REAL number, for example 2.4e-10, is placed in a FLOAT.
- A PACKED literal is placed in the smallest packed field that can contain it.

When the above assumptions are incorrect or are not desired, you can use a formatting directive to cause the desired output. The format of a formatting directive is: *form name(literal)*, where name is a field or literal or collect name from which the desired data type and length can be referred (it is not the position of the position of the referenced statement). The referenced name cannot have a length of asterisk. Here are some examples:

- To cause the value four to become a short: FIELDS(4) where FIELDS has previously been defined as a short.
- To produce a double of four use: FIELDD(4) where FIELDD is defined as a double.
- If you have a field called FIELD40 that is defined as a CHAR with a length of 40, then FIELD40("This") produces a 40 byte blank padded result with "This" in the left bytes.

- If you have a previously defined field of six bytes PACKED data called FIELDP6, then FIELDP6(p'-123') expands what appears to be a two byte literal to six bytes.
- You can expand a data field using a literal as follows. If you have a field called CUSTOMER that is 40 bytes in the source record and you want to allow 45 bytes in the output record, use:

```
'SELECT xxx , yyy , customer,"      ",next_field, FROM (predicate) '
```

 where there are 5 blanks in the literal.
- You can expand a PACKED field by two bytes using:

```
'SELECT field1, field2, x'0000', packed_field '
```

How use of the Cobol parser affects output alignment

IBM File Export alignment rules come into play when creating output records. These rules cause data types to be aligned on their optimal hardware boundaries. Because COBOL's alignment rules differ from those of IBM File Export, it will be necessary for you to take the following steps to ensure that output records are written correctly.

- When creating output using the IBM File Export user interface under ISPF, it is recommended that, when moving data, fields be defined with a data type of CHAR.
- When creating JCL directly using ABXUTIL, it is recommended that you define your output using the OFIELD declarative to avoid alignment problems.

Part 7. Appendixes

Appendix A. DDnames used by IBM File Export for z/OS

This section lists the reserved ddnames that are used by IBM File Export for z/OS and explains their use. In some cases, the function of these data sets differs depending on the type of data that you are working with. These differences are explained in this section.

ABXCTL

This dd is used to specify data gathering and manipulation instructions. It defines the data set containing the IBM File Export for z/OS control statements. This data set can reside on a DASD or TAPE device, or be routed through the input stream. The DCB attributes of this data set are RECFM=FB and LRECL=80. There is no variation in the function of this data set based on the type of data that is being processed.

ABXCNTLI

DB2:

This dd is the input control file that is used when processing DB2 data. It is populated by the IBM File Export for z/OS ISPF User Interface.

IMS:

This data set is not used when working with IMS data.

IMS Fast Path:

This data set is not used when working with IMS Fast Path data.

VSAM:

This data set is not used when working with VSAM data.

Sequential:

This data set is not used when working with sequential data.

ABXCNTLO

DB2:

This dd is a control file that is used when processing DB2 output data. It is populated by the contents of the ABXCNTLI data set and augmented by ABXUTIL and subsequently used by the DB2 LOAD EXECUTOR.

IMS:

This data set is not used when working with IMS data.

IMS Fast Path:

This data set is not used when working with IMS Fast Path data.

VSAM:

This data set is not used when working with VSAM data.

Sequential:

This data set is not used when working with sequential data.

ABXDDL

This dd must point to a data set that is RECFM=FB, LRECL=80 and will contain the generated DDL if the Generate DDL option is in effect.

ABXIN

DB2:

This data set is not used when working with DB2 data.

IMS:

This dd defines the HD-UNLOAD input file created by the HD Reorganization Unload Utility (DFSURGU0). This data set can reside on a DASD or TAPE device. This data set is not required and is unused when the LOAD keyword is included in the list of OPTIONS parameters specified in the ABXCTL input stream.

IMS Fast Path:

This data set is not used when working with IMS Fast Path data.

VSAM:

This dd defines the primary input VSAM file which may be an ESDS (Entry Sequence Data Set), KSDS (Key Sequence Data Set), RRDS (Relative Record Data Set) or an VRRDS (Variable Relative Record Data Set). This data set must reside on a DASD device. This data set is not required and is unused when the LOAD keyword is included in the list of OPTIONS parameters specified in the ABXCTL input stream.

Sequential:

This dd defines an input sequential file. It defines the primary input for SEQUENTIAL processing mode. This data set can reside on a DASD or TAPE device. This data set is not required and is unused when the LOAD keyword is included in the list of OPTIONS parameters specified in the ABXCTL input stream.

ABXOUT**DB2:**

This data set is not used when working with DB2 data.

IMS:

This dd defines the output file that will be in HD-UNLOAD format. This data set may reside on a DASD or TAPE device.

IMS Fast Path:

This data set is not used when working with IMS Fast Path data.

VSAM:

This dd defines the VSAM output file. This data set can be a Key Sequenced Data Set (KSDS), an Entry Sequenced Data Set (ESDS), a Relative Record Data Set (RRDS) or a Variable Relative Record Data Set (VRRDS).

Sequential:

This dd defines the sequential output file (target). This data set can reside on a DASD or TAPE device. The DCB LRECL and RECFM attributes for this data set are user-dependent. You can preallocate the data set or specify the LRECL and RECFM for the data set in the JCL. You can also have the LRECL and RECFM defined in the input stream, but you must supply the output data sets' DCB attributes via preallocation or JCL specification.

ABXPRINT

This dd is the data set where a copy of the control syntax and utility messages that are generated by IBM File Export for z/OS when it runs. The data set can reside on DASD, TAPE, or printer, or can be routed through the output stream. The DCB parameters for this data set are RECFM= FBA and LRECL=121. The BLKSIZE parameter if specified, must be a multiple of 121. If this data set is not included in the job stream, it will be dynamically allocated and the output will be routed to the MSGCLASS specified on the JOBCARD as SYSOUT. There is no variation in the function of this data set based on the type of data that is being processed.

ABXSQL

This dd is used when producing SQL output. It has format LRECL=80, BLKSIZE=n*.80,RECFM=FB. This dd contains data that is in a format suitable for insertion into a DB2 database using a tool such as SPUFI or IBM DB2 Administration Tool for z/OS.

IMSRPT

DB2:

This data set is not used when working with DB2 data.

IMS:

This dd defines the Segment Statistics and Database Definition Report output data set. This data set can reside on DASD, TAPE or printer, or can be routed through the output stream. The DCB parameters for this data set are: RECFM=FBA and LRECL=121. The BLKSIZE parameter, if specified, must be a multiple of 121.

IMS Fast Path:

This dd defines the Segment Statistics and Database Definition Report output data set. This data set can reside on DASD, TAPE or printer, or can be routed through the output stream. The DCB parameters for this data set are: RECFM=FBA and LRECL=121. The BLKSIZE parameter, if specified, must be a multiple of 121.

VSAM:

This data set is not used when working with VSAM data.

Sequential:

This data set is not used when working with sequential data.

NEWIMS

DB2:

This data set is not used when working with DB2 data.

IMS:

This dd defines the library containing the DBD definition (DBDLIB member) of the database named using the NEWDBDNAME parameter. This dd is optional.

IMS Fast Path:

This data set is not used when working with IMS Fast Path data.

VSAM:

This data set is not used when working with VSAM data.

Sequential:

This data set is not used when working with sequential data.

OLDIMS

DB2:

This data set is not used when working with DB2 data.

IMS:

This dd defines the library containing the DBD definition (DBDLIB member) for the database named in DBDNAME parameter.

IMS Fast Path:

This data set is not used when working with IMS Fast Path data.

VSAM:

This data set is not used when working with VSAM data.

Sequential:

This data set is not used when working with sequential data.

STEPLIB**DB2:**

This dd points to the IBM File Export for z/OS program library.

IMS:

This dd points to the IBM File Export for z/OS program library.

IMS Fast Path:

This dd points to the IBM File Export for z/OS program library. It must also include the product program library for the IBM Fast Path Basic Tools library. You can include the libraries that contain any Segment Edit/Compression or Randomizer routines if the RMODLIB DD statement will not be used.

VSAM:

This dd points to the IBM File Export for z/OS program library.

Sequential:

This dd points to the IBM File Export for z/OS program library.

SYSABEND

This is an alternative dump data set that is used by the operating system to output DUMP information when a failure occurs.

SYSUDUMP

This is the optional dump data set that is used by the operating system to output dump information when a failure occurs. The output is normally routed to SYSOUT=*

TABL001 through TABL*nnn***DB2:**

These data sets contain the load image data for LOAD processing in the target system. The first defined table statement creates the images in the data set pointed to by the dd named TABL001. The *nnn*'th table uses TABL*nnn*.

IMS:

These data sets are not used when working with IMS data.

IMS Fast Path:

These data sets are not used when working with IMS Fast Path data.

VSAM:

These data sets are not used when working with VSAM data.

Sequential:

This data set is not used when working with sequential data.

External data sets required when working with IMS Fast Path data

The DDs described in this section are required when you are working with IMS Fast Path data. IBM File Export for z/OS calls the data sets that are a part of the IMS Fast Path Basic Tools utility programs, FABCUR6 and FABCUR7. The required data sets from these utilities are as follows:

DD Statements from FABCUR6:

The data sets for FABCUR6 are required when the LOAD keyword is specified in OPTIONS parameter that is contained in the ABXCTL dd.

ACBLIB

Defines the data sets containing the database DMB. This dd is not required if the MODSTAT or MODSTAT2 dd is supplied. Otherwise, this dd is required.

IMSACBA

This dd must be provided if the MODSTAT or MODSTAT2 dd statements are used. This dd defines the data sets containing the database DMB.

IMSACBB

This dd must be provided if the MODSTAT or MODSTAT2 DD statements are used. This dd defines the data sets containing the database DMB.

MODSTAT

This dd defines the MODSTAT data set. If this dd is supplied, the ACBLIB dd statement is not used and the IMSACBA and IMSACBB dd statements are required.

MODSTAT2

This dd defines the MODSTAT2 data set. If this dd is supplied, the ACBLIB dd statement is not used and the IMSACBA and IMSACBB dd statements are required.

RMODLIB

This dd defines the data sets containing any Segment Edit/Compression routines that may be needed to process segments. This dd can be omitted if the libraries containing the programs have been included in the JOBLIB/STEPLIB dd concatenation.

UR6CTL

This dd describes the control statement input stream for FABCUR6.

UR6PRINT

This dd describes the output data set containing messages issued by FABCUR6.

UR6AUDIT

This dd describes the output data set containing the FABCUR6 Audit Report.

UR6DBDFN

This dd describes the output data set containing the database DMB information that will be used the DEDB Reload program, FABCUR3.

DURDzzzO

This dd describes the output data sets that contain processed segment data.

XDzzzzzO

This dd describes the output data sets that contain processed segment data.

DURDzzzE

This dd describes a secondary copy of the DURDzzzO data set.

XDzzzzzE

This dd describes a secondary copy of the XDzzzzzO data set.

DURSzzzO

This dd describes a data set that contains SORT control statements that may be required to correctly place the segment data in RAP/RBA order for reload processing.

XszzzzzO

This dd describes a data set that contains SORT control statements that may be required to correctly place the segment data in RAP/RBA order for reload processing.

DD Statements from FABCUR7:

The data sets for FABCUR7 are not required when the LOAD Keyword is specified in OPTIONS parameter that is contained in the ABXCTL dd.

UR7PRINT

This dd describes the output data set that contains messages produced by FABCUR7.

UR7AUDIT

This DD describes the output data set that contains the FABCUR7 Audit Control report.

RMODLIB

This dd defines the data sets containing any Segment Edit/Compression routines that may be needed to process segments. This DD can be omitted if the libraries containing the programs have been included in the JOBLIB/STEPLIB dd concatenation.

UR7CTL

This dd describes the control statement input stream for FABCUR7.

UR7DBDFN

This dd describes the database DMB data set created by the UNLOAD process (FABCUR1).

UR7DATA

This dd describes the input data sets containing the segment data created by the UNLOAD process.

Appendix B. Installation Verification Programs

This appendix provides information about the Installation Verification Programs (IVPs) provided with IBM File Export for z/OS. These programs can be found in SABXSAMP. Directions for running each IVP are provided as comments interspersed with the program's code.

Note: The sample code found in SABXSAMP may differ slightly from the printed samples found here. In the case that there is a discrepancy, the sample code in SABXSAMP should be considered the most current version.

IVP for DB2

SABXSAMP member ABXIVPJ3 contains the Installation Verification Program for DB2. The IVP can be used to determine that the DB2 connections are correct and that the plan for ABXUTIL is properly bound. The IVP does not continue to load the data and, as such, cannot be used to prove that the loader for DB2 has been properly bound.

The IVP consists of two steps:

- Step 1 creates a small file that contains a list of names to be used as an application-defined relationship (ADR) into the second step.
- Step 2 extracts data from DB2 and performs an extract match against the names produced in the first step. It then extracts from a child based on a parent-child relationship using CHAR to VARCHAR RI. Finally, THEN INSERT is used to create printable data to illustrate what the extracted data looks like.

Before running the program:

- Change the job card and the STEPLIB statements to point to the data sets where ABXUTIL and the DB2 access routines reside. Two occurrences each of the following statements must be changed:

```
//STEPLIB DD DISP=SHR, DSN=ABX.V1R1.LOAD  
          DD DISP=SHR, DSN=DSN.V610.SDSNLOAD
```

- Change the SOURCEDB2 and TARGETDB2 statements to correspond to one of your DB2 systems and reflecting the changes you previously made in the STEPLIB statements. Two occurrences each of the following statements must be changed:

```
SOURCEDB2(R61A);  
TARGETDB2(R61A);
```

All of the change points in the JCL are marked with the word 'CHANGE.'

IVP for IMS

There are no special installation or customization considerations for IMS that are not handled by the Sequential or VSAM IVP jobs. For information, refer to "IVP for VSAM" on page 274 and "IVP for sequential files" on page 274.

IVP for VSAM

SABXSAMP member ABXIVPJ2 contains the Installation Verification Program for VSAM files. This job consists of five steps:

- Step 1 deletes any existing output files
- Step 2 populates a VSAM ESDS
- Step 3 reads and writes VSAM data using ABXUTIL
- Step 4 extracts data from the new VSAM ESDS
- Step 5 compares the input data to the resulting output data

Each of these steps should produce return code 0 when run. Before running the program:

- Change all occurrences of `your.product.loadlib` to the data set name that contains the IBM File Export for z/OS product executables (load modules).
- Change all occurrences of `your.product.SABXSAMP(ABXIVPIN)` to the data set name of the IBM File Export for z/OS SABXSAMP that you installed. `ABXIVPIN` contains randomly generated hexadecimal data. Alternatively, you can use a sequential data set of your own choosing as input.
- Change all occurrences of `your.output.dataset.ps` to the data set name that you want to use as the sequential output data set that will hold the extracted data from the new VSAM data set. This data set's DCB attributes have been set to match those of the IBM File Export IVP process input data set that may be used as described above. If an alternate input data set is used to populate the VSAM data set, the DCB attributes for this file must be changed accordingly.
- Change all occurrences of `your.input.dataset.vsam` to the data set name that you want to use as the base VSAM data set that is to be populated and used as input to the IBM File Export process. You must supply a `VOLUME` parameter for this data set and, if you are using a different input sequential data set than that which was supplied in SABXSAMP, you may also need to change the `SPACE` parameter.
- Change all occurrences of `your.output.dataset.vsam` to the data set name that you want to use as the new VSAM data set that is to be populated by the IBM File Export process. You must supply a `VOLUME` parameter for this data set and, if you are using a different input sequential data set than that which was supplied in SABXSAMP, you may also need to change the `SPACE` parameter.

IVP for sequential files

SABXSAMP member ABXIVPJ1 contains the Installation Verification Program for sequential files. This job consists of three steps:

- Step 1 deletes any existing output files
- Step 2 reads and writes sequential data using ABXUTIL
- Step 3 compares the input data to the resulting output data

Each of these steps should produce return code 0 when run. Before running the program:

- Change all occurrences of `your.product.loadlib` to the data set name that contains the IBM File Export for z/OS product executables (load modules).
- Change all occurrences of `your.output.dataset` to the data set name that you want to use as the output data set.
- Change all occurrences of `your.product.SABXSAMP(ABXIVPIN)` to the data set name of the IBM File Export for z/OS SABXSAMP that you installed. `ABXIVPIN`

contains randomly generated hexadecimal data. Alternatively, you can use a sequential data set of your own choosing as input.

Appendix C. IBM File Export Global Options

This appendix describes the parameters that may be specified when using the IBM File Export Global options macro ABXMGBL0 and their defaults.

Parameters

ABXMGBL0	AUDIT=NO,	x	
	SMFNO=,		x
	INDDN=ABXIN,		x
	OTDDN=ABXOUT,		x
	PTDDN=ABXPRINT,		x
	SINDDN=ABXCTL,		x
	WUNIT=SYSDA,		x
	QUIET=NO,		x
	FIELDS=YES,		x
	DECSEP=DECIMAL,		x
	FILL=BLANKS,		x
	UXITACT=NO,		x
	UXITNM=,		

AUDIT NO | YES | FORCE:

NO This is the default. This parameter indicates that SMF audit records are not to be written to the system SMF data sets. This value may be overridden by including the AUDIT(); keyword in the ABXUTIL control statement stream.

YES This parameter indicates that SMF audit records are to be written to the system SMF data sets. If, for some reason, the request to write the SMF records fails, ABXUTIL issues a message indicating the reason for the failure, dynamically changes the AUDIT option to NO, and continues processing the input and output data without an audit trail. This value may be overridden by including the AUDIT(); keyword in the ABXUTIL control statement stream.

FORCE

This parameter indicates that SMF audit records are to be written to the system SMF data sets. If, for some reason, the request to write the SMF records fails, ABXUTIL issues a message indicating the reason for the failure, dynamically changes the AUDIT option to NO, and terminates processing. This value cannot be overridden by including the AUDIT(); keyword in the ABXUTIL control statement stream.

SMFNO 128 -- 255:

There is no default value for this parameter. It must be included when the AUDIT parameter has a value of YES or FORCE,. This parameter indicates the SMF record number that should be used by ABXUTIL when writing records to the system SMF data sets. The value should be determined after consulting with the site system programmer to avoid number conflicts with other program products.

INDDN ddname:

This parameter indicates the ddname to be used to define the input data set when using processing types VSAM, SEQUENTIAL, or IMS. The default value is ABXIN. Any 1- to 8-character string may be used. The string must follow MVS ddname naming rules and must not conflict with other specified ddnames. This value cannot be overridden using the ABXUTIL control statements. (See Appendix A, "DDnames used by IBM File Export for z/OS," on page 265 for a list of ddnames used by IBM File Export.

OTDDN ddname:

This parameter indicates the ddname to be used to define the output data set when using processing types VSAM, SEQUENTIAL, or IMS. The default value is ABXOUT. Any 1- to 8-character string may be used. The string must follow MVS ddname naming rules and must not conflict with other specified ddnames. This value cannot be overridden using the ABXUTIL control statements. (See Appendix A, "DDnames used by IBM File Export for z/OS," on page 265 for a list of ddnames used by IBM File Export.

PTDDN ddname:

This parameter indicates the ddname to be used to define the message print data set when using all processing types. The default value is ABXPRINT. Any 1- to 8-character string may be used. The string must follow MVS ddname naming rules and must not conflict with other specified ddnames. The value cannot be SYSPRINT. This value cannot be overridden using the ABXUTIL control statements. (See Appendix A, "DDnames used by IBM File Export for z/OS," on page 265 for a list of ddnames used by IBM File Export.

SINDDN ddname:

This parameter indicates the ddname to be used to define the ABXUTIL control statement input data set when using all processing types. The default value is ABXCTL. Any 1- to 8-character string may be used. The string must follow MVS ddname naming rules and must not conflict with other specified ddnames. The value cannot be SYSIN. This value cannot be overridden using the ABXUTIL control statements. (See Appendix A, "DDnames used by IBM File Export for z/OS," on page 265 for a list of ddnames used by IBM File Export.

WUNIT unit name:

This parameter indicates the esoteric name of the device that should be used when ABXUTIL needs to dynamically allocate new or temporary data sets. It is valid for all processing types. The default value is SYSDA. This value may be overridden by specifying the WUNIT (); keyword using ABXUTIL control statements.

QUIET NO|YES:

This parameter indicates whether the informational messages produced by ABXUTIL should be reduced. The default value is NO and is the equivalent of supplying the VERBOSE parameter on the OPTIONS statement of the ABXUTIL control statement stream. Specifying QUIET=YES results in all the suppression of all informational messages and is the equivalent of supplying the QUIET parameter on the OPTIONS statement in the ABXUTIL control statement stream.

FIELDS YES|NO:

This parameter indicates whether the informational messages describing the fields contained in the IMS DBD segment definitions should be produced by produced by ABXUTIL when processing IMS databases using the IMS, DEDB, DLI, DBB, or BMP processing modes. The default value is YES and is the equivalent of supplying the

FIELDS parameter on the OPTIONS statement in the ABXUTIL control statement stream. Specifying FIELDS=NO results in suppression of all informational messages about fields in the IMS segments. It is the equivalent of supplying the NOFIELDS parameter on the OPTIONS statement in the ABXUTIL control statement stream.

DECSEP DECIMAL|COMMA:

This parameter indicates whether a decimal point or a comma is to be used in real numbers. This parameter is valid for all processing types. The default value is DECIMAL and is the equivalent of supplying the DECIMAL parameter on the OPTIONS statement in the ABXUTIL control statement stream. Specifying DECSEP=COMMA results in a comma (x'6B') being used as the delimiter and is the equivalent of supplying the COMMA parameter on the OPTIONS statement in the ABXUTIL control statement stream.

FILL BLANKS|ZEROS|hex value(00-FF):

This parameter indicates the character to be used as the pad or fill character when format changes to an output record require the inclusion of filler or slack bytes, either to provide correct alignment or to complete a request to lengthen the record. This parameter is valid for all processing types. The default value of BLANKS will cause ABXUTIL to use blanks (x'40') as the fill character. A value of ZEROS will cause ABXUTIL to use zeros (x'F0') as the fill character. If a hex value is supplied, such as FILL=00, ABXUTIL will use that value as the fill character. This value may be overridden by specifying the FILLER(); keyword in the ABXUTIL control statement stream.

UXITACT NO|YES:

This parameter indicates whether the user-supplied I/O exit program should be invoked for each record that is read or will be written. The default value is NO. This parameter is valid for all processing types. It may be overridden by specifying the EXITACTIVE(); keyword in the ABXUTIL control statement stream.

UXITNM program name:

This parameter specifies the name of the program to be invoked by ABXUTIL for every record that is read or will be written. There is no default value. This parameter is valid for all processing types. It must be specified if UXITACT=YES is specified. The parameter may be overridden by specifying the EXITNAME(); keyword in the ABXUTIL control statement stream.

Appendix D. IBM File Export DB2 Options

This appendix describes the parameters that may be specified when using the IBM File Export DB2 options macro ABXMDB20 and their defaults.

Parameters

ABXMDB20TYPE=	x
SSID=,	x
COMMENT=,	x
LOADLIB=,	x
UIPLAN=ABXDBUT,	x
BATCHPLAN=ABXUTIL,	x
LOADPLAN=ABXD2LEX	

TYPE START | ENTRY | FINAL:

START

This parameter indicates that the parameter build process is to start. It must be specified in the first occurrence of the ABXMDB20 macro. Only one occurrence may be present, and no other parameters are valid.

ENTRY

This parameter indicates that an entry is to be built for a specific DB2 subsystem.

FINAL

This parameter indicates that the parameter build process is complete. It must be specified in the last occurrence of the ABXMDB20 macro. Only one occurrence may be present, and no other parameters are valid.

SSID DB2 subsystem ID:

This parameter is only valid when specified along with the TYPE=ENTRY keyword. It identifies the DB2 subsystem name this entry relates to. There is no default value.

COMMENT 1-40 character bytes:

This parameter is only valid when specified along with the TYPE=ENTRY keyword. It allows the user to place a comment of up to 40 bytes long in the entry, and may be used to describe the subsystem. This comment is displayed when the DB2 Option block contents are displayed by the IBM File Export user interface.

UIPLAN plan name:

This parameter is only valid when specified along with the TYPE=ENTRY keyword. It specifies the DB2 plan name that the IBM File Export user interface will use to ??? the default value is ABXDBUT. This value may be overridden by specifying the PLAN() keyword on the OPTIONS statement in the ABXUTIL control statement stream.

BATCHPLAN plan name:

This parameter is only valid when specified along with the TYPE=ENTRY keyword. It specifies the DB2 plan name that the program IBM File Export user interface will use to access DB2 data ??? The default value is ABXUTIL. This value cannot be overridden using ABXUTIL control statements.

LOADPLAN plan name:

This parameter is only valid when specified along with the TYPE=ENTRY keyword. It specifies the DB2 plan name that the DB2 Load Executor program, ABXD2LEX, will use to write to DB2 tables. The default value is ABXD2LEX. This value cannot be overridden using ABXUTIL control statements.

Appendix E. IBM File Export IMS Options

This appendix describes the parameters that may be specified when using the IBM File Export IMS options macro ABXMIMS0 and their defaults.

Parameters

ABXMIMS0	TYPE=	X
IMSID=,		X
COMMENT=		X
ALTID=,		X
SSM=,		X
IMSPLEX=,		X
ALTRRC00=,		X
DBRC=,		X
IRLM=,		X
AGN=,		X
DLTDUPS=N,		X
IRLMNAME=,		X
GSGNAME=,		X
DPSBPFX=N,		X
DPSBSFX=,		X
DPSBSFX=,		X
LOCKMAX=,		X
DPSBMAC=,		X
DBDLIB=,		X
DOPTLIB=,		X
RESLIB1=,		X
RESLIB2=,		X
RESLIB3=,		X
DFSVSAMP=,		X
CPUTIME=,		X
DIRCA=,		X
NBA=,		X
OBA=,		X
OPT=,		X

PARDLI=,	x
STIMER=,	x
BK0=,	x
BUF=,	x
EXCPVR=,	x
FMT0=,	x
MON=,	x
SWAP=,	x
PROCOPT=	

TYPE START | ENTRY | FINAL:

START

This parameter indicates that the parameter build process is to start. It must be specified in the first occurrence of the ABXMIMS0 macro. Only one occurrence may be present, and no other parameters are valid.

ENTRY

This parameter indicates that an entry is to be built for a specific IMS subsystem.

FINAL

This parameter indicates that the parameter build process is complete. It must be specified in the last occurrence of the ABXMIMS0 macro. Only one occurrence may be present, and no other parameters are valid.

IMSID IMS subsystem ID:

This parameter is only valid when specified along with the TYPE=ENTRY keyword. It identifies the IMS subsystem name this entry relates to. There is no default value. This value is included in the IMS region controller parameter list in the appropriate position.

COMMENT 1--40 character bytes:

This parameter is only valid when specified along with the TYPE=ENTRY keyword. It allows the user to place a comment of up to 40 bytes long in the entry, and may be used to describe the subsystem. This comment is displayed when the IMS Option block contents are displayed by the IBM File Export user interface.

ALTID 1--4 character bytes:

This parameter specifies an alternate IMSID that a BMP region will attempt to connect to if the primary subsystem ID (IMS ID) is not available for processing. This parameter is only valid when specified along with the TYPE=ENTRY keyword. There is no default value. This value may be overridden by specifying the ALTID(); keyword in the ABXUTIL control statement stream. This parameter is used when processing in BMP mode. This parameter is passed to the IMS controller in the appropriate position of the input parameters. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the ALTID parameter.

SSM 1--4 character bytes:

This parameter specifies the suffix of a member name used to control access to external subsystems. This parameter is only valid when specified along with the TYPE=ENTRY keyword. There is no default value. This value may be overridden by specifying the SSM(); keyword in the ABXUTIL control statement stream. This region

controller parameter is used when processing in BMP mode. This parameter is included in the IMS region controller parameter list in the appropriate position. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the SSM parameter.

IMSPLEX 5-character name:

This parameter specifies the 5-character value that is passed to the SCI registration exit, DSPSCIX0. This parameter is only valid when specified along with the TYPE=ENTRY keyword. There is no default value. This value may be overridden by specifying the IMSPLEX(); keyword in the ABXUTIL control statement stream. This region controller parameter is used when processing in DLI and DBB mode. This parameter is included in the IMS region controller parameter list in the appropriate position. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the IMSPLEX parameter.

ALTRRC00 1--8 character program name:

This parameter allows you to execute an alternate IMS region controller program to DFSRRC00. This parameter is only valid when specified along with the TYPE=ENTRY keyword. There is no default value. This value may be overridden by specifying the ALTRRC00(); keyword in the ABXUTIL control statement stream. This parameter is used when processing in DLI, DBB, and BMP mode.

DBRC NO|YES|blank:

This parameter specifies whether DBRC should be used for this execution. Valid values are Y or YES, indicating that DBRC should be used N or NO, indicating that DBRC should not be used, or blank, indicating that the DBRC= value specified in the IMSCTL macro during IMS system definition is to be used. . This parameter is only valid when specified along with the TYPE=ENTRY keyword. The default is blank. This value may be overridden by specifying the DBRC(); keyword in the ABXUTIL control statement stream. This region controller parameter is used when processing in DLI or DBB mode. This parameter is included in the IMS region controller parameter list in the appropriate position. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the DBRC parameter.

IRLM NO|YES|blank:

This parameter specifies whether IRLM should be used for this execution. Valid values are Y or YES, indicating that IRLM should be used, N or NO, indicating that IRLM should not be used, or blank, indicating that the IRLM= value specified in the IMSCTL macro during IMS system definition is to be used. This parameter is only valid when specified along with the TYPE=ENTRY keyword. The default is blank. This value may be overridden by specifying the IRLM(); keyword in the ABXUTIL control statement stream. This region controller parameter is used when processing in DLI or DBB mode. This parameter is included in the IMS region controller parameter list in the appropriate position. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the DBRC parameter.

AGN 1--8 character bytes:

This parameter specifies the Application Group Name used for interregion communication security. This parameter is only valid when specified along with the TYPE=ENTRY keyword. The default value is blank. This value may be overridden by specifying the AGN(); keyword in the ABXUTIL control statement stream. This parameter is used when processing in BMP mode. This parameter is included in the IMS region controller parameter list in the appropriate position. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the AGN parameter.

DLTDUPS NO|YES:

This parameter specifies whether ABXUTIL should automatically delete any duplicate segments that are found in an IMS database when writing to that database. Valid values are Y or YES, indicating that duplicate segments should be deleted, or the default value, N or NO, indicating that duplicate segments should not be deleted. This parameter is only valid when specified along with the TYPE=ENTRY keyword. The default is blank. This value may be overridden by specifying the DLTDUPS() keyword in the ABXUTIL control statement stream. This parameter is used when processing in DLI, DBB, or BMP mode. This parameter is included in the IMS region controller parameter list in the appropriate position.

IRLMNAME 1--8 character bytes:

This parameter specifies the name of the IRLM subsystem that should be used if IRLM is active for this execution. This parameter is only valid when specified along with the TYPE=ENTRY keyword. There is no default value. This value may be overridden by specifying the IRLMNAME() keyword in the ABXUTIL control statement stream. This region controller parameter is used when processing in DLI and DBB mode. This parameter is included in the IMS region controller parameter list in the appropriate position. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the IRLMNAME parameter.

GSGNAME 1--8 character bytes:

This parameter specifies the Global Services Group name that should be used for this execution. This parameter is only valid when specified along with the TYPE=ENTRY keyword. There is no default value. This value may be overridden by specifying the GSGNAME() keyword in the ABXUTIL control statement stream. This region controller parameter is used when processing in DLI and DBB mode. This parameter is included in the IMS region controller parameter list in the appropriate position. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the GSGNAME parameter.

DYNPSB YES|NO:

This parameter specifies whether the Dynamic PSB feature should be used for this execution. Valid values are Y or YES, indicating that Dynamic PSB feature should be used, or N or NO, indicating that the Dynamic PSB feature should not be used. Valid when specified along with the TYPE=ENTRY keyword. The default is NO. This value may be overridden by specifying the DYNPSB() keyword in the ABXUTIL control statement stream. This parameter is used when processing in DLI, DBB, or BMP mode.

DPSBPFX 1--6 characters:

This parameter specifies the prefix to be used for the names of the PSBs that are generated by the Dynamic PSB feature. This 6-character prefix together with a 2-character suffix comprise the PSB's name. The 2-character suffix may be any 2-byte value ranging from 01 through the numeric value you specify using the DPSBSFX keyword. This parameter is only valid when specified along with the TYPE=ENTRY keyword. This value may be overridden by specifying the DPSBPFX() keyword in the ABXUTIL control statement stream. This parameter is used when processing in DLI, DBB, or BMP mode. This parameter is included in the IMS region controller parameter list in the appropriate position. To learn more about dynamic PSBs, see the chapter so entitled in the *IBM File Export for z/OS: User's Guide*, as well as the section entitled "Security" on page 14 of this manual.

DPSBSFX 2-character numeric value from 01 -99 :

This parameter specifies the suffix to be used for the names of the PSBs that are generated by the Dynamic PSB feature. This 2-character suffix together with a 6-character prefix specified using the DPSBPFX keyword comprise the PSB's name. The 2-character suffix may be any 2-byte value ranging from 01 through 99. This parameter is only valid when specified along with the TYPE=ENTRY keyword. This value may be overridden by specifying the DPSBSFX(); keyword in the ABXUTIL control statement stream. This parameter is used when processing in DLI, DBB, or BMP mode. This parameter is included in the IMS region controller parameter list in the appropriate position. To learn more about dynamic PSBs, see the chapter so entitled in the *IBM File Export for z/OS: User's Guide*, as well as the section entitled "Security" on page 14 of this manual.

LOCKMAX 1--32767 (1 to 5 numeric digits):

This parameter specifies the value that may override a LOCKMAX value that was specified in a PSBGEN. This parameter is only valid when specified along with the TYPE=ENTRY keyword. This value may be overridden by specifying the LOCKMAX(); keyword in the ABXUTIL control statement stream. This parameter is used when processing in DLI, DBB, or BMP mode. This parameter is included in the IMS region controller parameter list in the appropriate position. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the GSGNAME parameter.

DPSBMAC 1--44 characters:

This parameter specifies the data set name of the partitioned data set that contains the IMS macros used to generate PSBs when using the Dynamic PSB feature.. This value may be overridden by specifying the DPSBMAC(); keyword in the ABXUTIL control statement stream. This parameter is used when processing in DLI, DBB, or BMP mode.

DBDLIB 1--44 characters:

This parameter specifies the data set name of the partitioned data set that contains the DBDLIB members associated with the IMS subsystem. This value may be overridden by specifying the DBDLIB(); keyword in the ABXUTIL control statement stream. This parameter is used when processing in DLI, DBB, or BMP regions using the Dynamic PSB feature.

DOPTLIB 1-44 characters:

This parameter specifies the data set name of the partitioned data set that will contain the ACBLIB members that are created by IBM File Export when the Dynamic PSB feature is used while processing in BMP mode. This value may be overridden by specifying the DOPTLIB(); keyword in the ABXUTIL control statement stream. The library specified must follow the rules for DOPT ACBLIB data sets as described in the *IMS Installation Volume 2: System Definition and Tailoring* manual and the *IMS Utilities : System* manual.

RESLIB1 1--44 characters:

This parameter specifies the name of the SVC data set that contains the IMS nucleus for the IMS subsystem. This value may be overridden by specifying the RESLIB1(); keyword in the ABXUTIL control statement stream. This parameter is used when processing in DLI, DBB, or BMP regions using the Dynamic PSB feature.

RESLIB2 1--44 characters:

This parameter specifies the name of the data set that contains any IMS dynamic allocation members for the IMS subsystem. This value may be overridden by

specifying the RESLIB2(); keyword in the ABXUTIL control statement stream. This parameter is used when processing in DLI, DBB, or BMP regions using the Dynamic PSB feature.

RESLIB3 1--44 characters:

This parameter specifies the name of the data set that contains any user programs or exits needed by IMS to access IMS databases. This value may be overridden by specifying the RESLIB3(); keyword in the ABXUTIL control statement stream. This parameter is used when processing in DLI, DBB, or BMP regions using the Dynamic PSB feature.

DFSVSAMP 1--44 characters:

This parameter specifies the name of the data set that contains the buffer and subpool information for databases. This data set name may also include a member name. This value may be overridden by specifying the DFSVSAMP(); keyword in the ABXUTIL control statement stream or by specifying a DFSVSAMP DD statement in the ABXUTIL step JCL. This parameter is used when processing in DLI, DBB, or BMP mode.

CPUTIME 1--4 numeric digits (0--1440):

This parameter specifies the maximum task time a BMP region may consume in minutes. Supported values are 0--1440 and blanks. This value may be overridden by specifying the CPUTIME(); keyword in the ABXUTIL control statement stream. This value is included in the IMS region controller parameter list in the appropriate position. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the CPUTIME parameter.

DIRCA 3 numeric digits (000--999):

This parameter specifies the number of 1K blocks of storage that is to be used as the interregional communications area. Supported values are from 000 through 999. This value may be overridden by specifying the DIRCA(); keyword in the ABXUTIL control statement stream. This value is included in the IMS region controller parameter list in the appropriate position. See the manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the DIRCA parameter.

NBA 1--4 numeric digits (0--9999):

This parameter specifies the number of IMS buffers that should be used when processing Fast Path databases. Supported values are 0 through 9999. This value may be overridden by specifying the NBA(); keyword in the ABXUTIL control statement stream. This parameter is used when processing in BMP mode. It is included in the IMS region controller parameter list in the appropriate position. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the NBA parameter.

OBA 1--4 numeric digits (1--9999):

This parameter specifies the number of IMS buffers that should be used as overflow when processing Fast Path databases should the NBA value prove insufficient. Supported values are 0 through 9999. This value may be overridden by specifying the OBA(); keyword in the ABXUTIL control statement stream. This parameter is used when processing in BMP mode. It is included in the IMS region controller parameter list in the appropriate position. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the OBA parameter.

OPT W|N|C:

This parameter specifies the the action a BMP region should take if the IMS subsystems described using the IMSID and ALTID keywords are not active. W indicates that the BMP region should wait for the IMS control region to become active. N indicates that the BMP region should request operator action. C indicates that the BMP region should terminate. This value may be overridden by specifying the OPT(); keyword in the ABXUTIL control statement stream. This parameter is used when processing in BMP mode. It is included in the IMS region controller parameter list in the appropriate position. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the OPT parameter.

PARDLI 0|1:

This parameter specifies whether DL/I access to the IMS databases should occur in the IMS control region (value 1) or the BMP region (value 0). This value may be overridden by specifying the PARDLI(); keyword in the ABXUTIL control statement stream. This parameter is used when processing in BMP mode. It is included in the IMS region controller parameter list in the appropriate position. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the PARDLI parameter.

STIMER 0|1|2|blank:

This parameter specifies whether processor time statistics are to be gathered. This value may be overridden by specifying the STIMER(); keyword in the ABXUTIL control statement stream. This parameter is used when processing in BMP mode. It is included in the IMS region controller parameter list in the appropriate position. The meaning of the parameter values is determined by the specification of other keyword parameters. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the STIMER parameter and the meaning of the various parameter values you may specify.

BKO YES|NO:

This parameter specifies whether (Y or YES) or not (N or NO) dynamic backout should occur if a pseudo abend occurs. This value may be overridden by specifying the BKO(); keyword in the ABXUTIL control statement stream. This parameter is used when processing in DLI or DBB mode. It is included in the IMS region controller parameter list in the appropriate position. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the BKO parameter.

BUF 1--3 digits:

This parameter specifies the number of 1K blocks used in calculating the size of OSAM buffer subpools. Valid value is from 000 through 999. This value may be overridden by specifying the BUF(); keyword in the ABXUTIL control statement stream. This parameter is used when processing in DLI and DBB mode. It is included in the IMS region controller parameter list in the appropriate position. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the BUF parameter.

EXCPVR 0|1:

This parameter specifies whether OSAM buffer pools should be page-fixed. A value of 1 indicates that the buffers should be page-fixed. A value of 0 indicates that they should not. This value may be overridden by specifying the EXCPVR(); keyword in the ABXUTIL control statement stream. This parameter is used when processing in DLI and DBB mode. It is included in the IMS region controller parameter list in the appropriate position. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the EXCPVR parameter.

FMTO 1 character:

This parameter specifies the format of any IMS dumps. This value may be overridden by specifying the FMTO(); keyword in the ABXUTIL control statement stream. This parameter is used when processing in DLI and DBB mode. Many values are supported. For information about the supported values, see the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*.

MON NO|YES:

This parameter specifies whether (YES) or not (NO) the IMS monitor should be active for this execution. This value may be overridden by specifying the MON(); keyword in the ABXUTIL control statement stream. This parameter is used when processing in DLI and DBB mode. It is included in the IMS region controller parameter list in the appropriate position. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the MON parameter.

SWAP NO|YES:

This parameter specifies whether (YES) or not (NO) the address space is swappable. This value may be overridden by specifying the SWAP(); keyword in the ABXUTIL control statement stream. This parameter is used when processing in DLI and DBB mode. It is included in the IMS region controller parameter list in the appropriate position. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the SWAP parameter.

PROCOPT G|GO:

This parameter specifies the IMS PROCOPT that should be used to read databases when using a dynamically built PSB. Valid options are G and GO. This value may be overridden by specifying the PROCOPT(); keyword in the ABXUTIL control statement stream. This parameter is used when processing in DLI, DBB, and BMP mode. See the IBM manual, *IMS Installation Volume 2: System Definition and Tailoring*, for more details about the PROCOPT parameter.

Understanding static and dynamic PSBs

Using IBM File Export to access IMS databases directly when processing DLI, DBB, or BMP regions requires the use of an IMS control block known as a PSB. A PSB (Program Specification Block) is used to define and control access to IMS databases by using one or more PCBs (Program Control Blocks) defined within the PSB. When used by an IMS DLI region, these PSBs usually reside in a partitioned data set, called a PSBLIB, in load module format. When used by IMS DBB and BMP regions, additional processing called an ACBGEN is executed and the PSB is written in a different format to a separate partitioned data set, which is known as an ACBLIB. For more information about the creation of PSBs, see the *IMS: Utilities Reference Manual*.

IBM File Export can use PSBs that have already been coded with the appropriate PCBs for the required databases. These are known as static PSBs. These static PSBs must reside in a PSBLIB when accessing IMS databases using a DLI region. For DBB regions, the PSB must reside in an ACBLIB. For BMP regions, the PSB must reside in an ACBLIB that is represented by a DD statement in the IMS control region, and, if applicable, in a DLI/SAS region. The PSB must also be defined to IMS during the IMS GEN process using the IMS APPLCTN macro. Only PSBs that have the LANG=ASSEM or LANG=COBOL optional keyword included in the PSBGEN macro statement may be used. PSBs that have this option specified as PL/I, PASCAL, JAVA, or blank should not be used.

For more information on how to define the PSB to IMS, see *IMS Installation Volume 2: System Definition and Tailoring*.

IBM File Export can optionally generate PSBs that contain all the PCBs needed to perform the task at hand. These dynamically-generated PSBs are known as dynamic PSBs. These dynamic PSBs are created at execution time and only exist for the duration of the job step that created them. For all region types, IBM File Export creates the PSB source statements to define PCBs with a read-only processing option (PROCOPT) for any databases that are to be read, and PCBs with a PROCOPT of ALL for databases that are to be written to. IBM File Export then dynamically invokes the assembler and linkage editor to create a PSB load module in a temporary partitioned data set (PDS).

The PSB in the temporary data set is used by IMS DLI region processing to gain access to the databases. For DBB and BMP regions, the ACBGEN process is then dynamically invoked and a PSB is created in ACBLIB format. In the case of a DBB region, this PSB is created in a temporary ACBLIB data set. In the case of a BMP region, this PSB is created in a predefined ACBLIB data set.

Using static PSBs

In order to use static PSBs, the dynamic PSB option must be OFF in the IMS Options Control Block, or you must have specified DYNPSB=N in the ABXUTIL Options control statement. You must also specify the following in the ABXUTIL control statement stream or by using the IMS Options Control Block:

- Specify the PSB name that you wish to use on the PSBNAME= keyword.
- Specify the PCB number in the PSB for the input database using the ABXUTIL control statement SOURCEDBD= or the INPCB#= parameter on the IMS Options Control Block.
- Specify the PCB number for the output database using the ABXUTIL control statement TARGETDBD= or the OUTPCB#= parameter on the IMS Options Control Block.

The static PSB to be used must reside in the appropriate data set with an appropriate DD statement referencing that data set. For example, for DLI regions, the PSB must reside in a valid PSBLIB data set, which is referenced by the IMS DD statement in the JCL. For DBB and BMP regions, the IMSACB DD statement or the MODSTAT/IMSACBA/IMSACBB combination of DD statements may be used to point to an ACBLIB data set. Only PSBs that have the LANG=ASSEM or LANG=COBOL optional keyword included in the PSBGEN macro statement may be used. PSBs that have this option specified as PL/I, PASCAL, JAVA, or blank should not be used.

See *IMS Installation Volume 2: System Definition and Tailoring* for details of DD statement requirements.

Detailed specifications for the ABXUTIL control statement syntax are found in the *IBM File Export for z/OS: Reference Manual* in the section entitled "Creating JCL Using Control Statements." Information about the IMS Options Control Block can be found in that manual in the section "Configuring IBM File Export to Process IMS Databases."

Using dynamic PSBs

In order to use IBM File Export's dynamic PSB function, the dynamic PSB option must be ON in the IMS Options Control Block, or you must have specified DYNPSB=Y in the ABXUTIL Options control statement. You must also specify the following by using the IMS Options Control Block:

- The dynamic PSB name prefix (DYNPPFX=)
- The dynamic PSB name suffix (DYNPSFX=)

- The IMS PSB Macro library dataset name (PSBMAC=)
- The IMS DBDLIB dataset (DBDLIB=)
- If the step is to execute as a BMP region, the DOPT ACBLIB dataset name (DOPT=)

Detailed specifications for the ABXUTIL control statement syntax are found in the *IBM File Export for z/OS: Reference Manual* in the section entitled "Creating JCL Using Control Statements." Information about the IMS Options Control Block can be found in that manual in the section "Configuring IBM File Export to Process IMS Databases."

Appendix F. Writing an I/O Exit

IBM File Export allows you to provide a user I/O exit to process records after they have been read or before they are written or both. Processing may include compression, encryption, or other nonstandard activities performed at the record level. While not intended to fully support encoded or compressed libraries, IBM File Export can access members of proprietary compressed libraries in PDS or other formats by means of the I/O compression exit, as long as the member name is known and is provided to IBM File Export.

Parameters are defined for the user I/O exit by using specifications in the IBM File Export global options (see Appendix C, "IBM File Export Global Options," on page 277), or by supplying specific control statements in program ABXUTIL.

Exit Protocol

The user I/O exit provides record level support. IBM File Export will perform all data I/O function and allows the exit to process the record after it is read or before it is written. The exit supports the following protocol:

Initialization

The exit indicates that the specified data set is to be processed by the exit and returns information about supported functions (once per data set).

Open IBM File Export opens the data set and provides vital information about the data. This call is performed only when processing sequential or VSAM data, not IMS databases.

A series of record access calls implementing the following functions:

- Record read
- Record write

The record to be processed is provided in an input buffer. The exit should process the data and return it in the output buffer while setting the output (record) length. A length of zero for the output record indicates that the processing did not change the record and the input record can be used "as is". At any time during read/write processing, the exit may indicate that it should no longer process the current data set. IBM File Export will stop invoking the exit for read/write processing and continues processing without the involvement of the exit. The Close and Terminate calls will still follow. Error codes may be set and messages returned by the exit. These messages will be displayed or printed for the user. Some codes will immediately terminate the current function while some will allow for continued processing. The messages will be displayed regardless of the return code every time that a non-blank message is returned by the exit.

Close

IBM File Export closes the data set after returning from the exit. This call is only performed when processing sequential or VSAM data, not IMS databases.

Termination

The exit performs any required "clean-up" processing.

Writing your exit

IBM File Export does not supply a default I/O exit. If you plan to use an I/O exit to process your data, you must provide one or more for your installation. You must also instruct IBM File Export to call the exit by supplying the exit name in the UXITNAM parameter and a value of **YES** in the UXITACT parameter of the IBM File Export Global Block. For information about providing parameters to the Global Block, see Appendix C, "IBM File Export Global Options," on page 277.

Alternatively, you can supply the exit name as a value of the EXITNAME keyword and the value of **YES** to the EXITACTIVE keyword in the control statement input stream to program ABXUTIL. For information on supplying the appropriate keywords to invoke the user exit, see Chapter 10, "Creating JCL Using Control Statements," on page 203.

Any exit you provide must be in the form of a load module, stored in any load library available to IBM File Export and accessed by a STEPLIB DD statement, or in LINKLIST or LPALIST. If IBM File Export attempts to load an exit using the Language Environment (LE) macro CEEFETCH, and if the exit is not found, you will receive an error message.

You can write your own exit in any high-level language such as COBOL, PL/I, or High Level Assembler. Sample exits for COBOL, PL/I, and HLASM are provided to help you do this. They are distributed in the ABX.SABXSAMP sample data set and are named as follows:

- **ABXUIOXA** -- Sample HLASM exit
- **ABXUIOXB** -- Sample COBOL exit
- **ABXUIOXP** -- Sample PL/I exit

Copybooks that provide the I/O exit control blocks for COBOL, PL/I and HLASM are distributed in the ABX.SABXSAMP sample data set and are named as follows:

- **ABXIOXA** -- Control block for HLASM programs
- **ABXIOXB** -- Control block for COBOL programs
- **ABXIOXP** -- Control block for PL/I programs

Programming your I/O exit

When programming your I/O exit, you should ensure that:

- The exit operates in the IBM Language Environment (LE). For additional information, refer to the IBM z/OS Language Environment: *Programming Guide*.
- The I/O exit executes in a non-XPLINK environment
- The I/O exit must be AMODE(31)
- The exit should be re-entrant

The following table provides information about the structure of the exit control block:

Data Name	Data Type	Set by:	Value
Eyecatcher	CL8	IBM File Export Initialization	"ABXUIOX"
Interface Version	F	IBM File Export Initialization	1 Initial Value

Data Name	Data Type	Set by:	Value
Level Support	F	EXIT Initialization	0 Process without exit 1 I/O performed by IBM File Export 2 Reserved for future use
Function code	F	IBM File Export: every call to exit	1 Initialize once per data set 2 Terminate once per data set 3 Open, once per data set, never with IMS 4 Close, once per data set, never with IMS 5 Read, required 6 Write, required 7 Reserved for future use 8 Reserved for future use 9 Reserved for future use
Exit return code	F	EXIT: Every call	0 OK, default. See Note 1 4 Warning 8 Error; terminate function 12 Severe error; terminate function 16 Fatal error; terminate function
Warning code	F	EXIT: Every call	0 Normal warning, default. See Note 1 1 Reserved for future use 2 Complete, continue processing without exit.
Reserved	XL6		Reserved
Functions supported by exit	X	EXIT: Initialization	1 Read, required 2 Write, required 3 Read and write, default 32 Reserved for future use 64 Reserved for future use 128 Reserved for future use
Operating Mode	X	IBM File Export Initialization	1 Reserved 8 Reserved 16 Reserved 32 Reserved 128 Batch, without TSO
DDNAME	CL8	IBM File Export Initialization	DDname allocated by IBM File Export. Not used when processing IMS databases
DSN	CL44	IBM File Export Initialization	Fully qualified DSN. Not used when processing IMS databases
Member Name	CL8	IBM File Export Initialization	Member name, only if data set is opened as sequential, not PO. Not used when processing IMS databases.
VOLSER	CL6	IBM File Export Initialization	First VOLSER of the data set, set to blank if not used. Not used when processing IMS databases.
DSORG	2X	IBM File Export OPEN	Data set organization: 1st byte: 1 Reserved 2 PDS(E) 32 DA 64 PS 128 Reserved 2nd byte: 8 VSAM 16 Reserved

Data Name	Data Type	Set by:	Value
VSAM catalog entry	X	IBM File Export OPEN	8 Reserved 16 Reserved 32 Reserved 64 Reserved 128 Cluster
VSAM type information	X	IBM File Export Open	4 Reserved 8 Reserved 16 VRDS 32 RRDS 64 ESDS 128 KSDS
RECFM	XL1	IBM File Export Open	x'C0' Undefined x'80' Fixed x'40' Variable x'20' Track Overflow x'10' Blocked x'08' Spanned x'04' ASA character x'02' Machine character
ALLOC Mode	XL1	IBM File Export Open	128 Temporary 32 SHR 8 (NEW,CATLG) 4 NEW 2 MOD 1 OLD
OPEN Mode	XL1	IBM File Export Open	8 VSAM initial LOAD 2 OUTPUT 1 INPUT
Reserved	XL11		Reserved
LRECL	F	IBM File Export Open	Record length (Non-VSAM)
BLKSIZE	F	IBM File Export OPEN	Block Size (Non-VSAM)
MLRECL	F	IBM File Export OPEN	Maximum VSAM record length
CISIZE	F	IBM File Export OPEN	CI size (VSAM)
KEYLEN	F	IBM File Export OPEN	Key length
RKP	F	IBM File Export OPEN	Relative key position
RBA	XL8	IBM File Export OPEN	XRBA (VSAM). Not used when processing IMS databases
SLOT	F	IBM File Export OPEN	Slot number (VVRDS or RRDS). Not used when processing IMS databases
Reserved	2F		Reserved
Current input record address	F	IBM File Export All I/O	I/O area address passed to the exit (size at least as big as the maximum record length) for the output record, holds the record to be processed by the exit.

Data Name	Data Type	Set by:	Value
Current input record length	F	IBM File Export All I/O	Length of record passed to exit
Current output record address	F	IBM File Export All I/O	I/O area passed to the exit. Used by the exit to return a modified output record.
Current output record length	F	IBM File Export All I/O	Length of record passed from the exit. If zero, the input records is assumed to be unchanged and the output record will be ignored.
Reserved	XL92		Reserved
Message	CL80	EXIT All	Message-- can be provided for output for each operation
User scratchpad area	CL32	EXIT All	Maintained by the exit from INIT through TERM
Reserved	XL128		Reserved
IMS type	CL4	IBM File Export Initialization	IMS processing type: IMS - IMS HD-UNLOAD DEDB - IMS Fast Path UNLOAD DLI - IMS offline direct database access DBB - IMS offline direct database access BMP - IMS online direct database access FEUN - IBM File Export UNLOAD format
IMS database name	CL8	IBM File Export All	IMS database name to be processed
IMS SSID	CL4	IBM File Export All	IMS subsystem ID being processed
Segment name	CL8	IBM File Export All I/O	Name of segment being passed to exit
Segment code	H	IBM File Export All I/O	Segment code of segment being passed to exit
Segment hierarchical level	H	IBM File Export All I/O	Hierarchical level of segment being passed to exit
Segment length indicator	CLI	IBM File Export All I/O	Segment type of segment being passed to exit: V - Variable length segment F - Fixed length segment
Reserved	XL27		Reserved

Notes:

1. Zero is the default. This code is initialized to zero on every call from IBM File Export to the exit.

Using the I/O exit control block

The I/O exit control block is the single parameter passed with each call from IBM File Export to the I/O exit. It is also used to return data from the I/O exit to IBM File Export.

Function codes and the flow of I/O exit processing

The I/O exit control block contains a Function Code field. The field is set by IBM File Export before the I/O exit is called, and indicates the type of I/O function that is being performed. The I/O exit is given the opportunity to perform actions at the points named by the function codes during IBM File Export processing. The function codes are called Initialization, Termination, Open, Close, Read, and Write. They are also discussed in the section entitled "Exit functions".

The I/O exit gets control as follows:

- **Open:** After the "real" data set Open
- **Close:** Before the "real" Close
- **Read:** Immediately after a Read from DASD (before IBM File Export examines or manipulates the data)
- **Write:** Immediately before a Write to DASD (after IBM File Export examines or manipulates the data)

The Function Code is the key input that your I/O exit should use to determine what action to take. The following sections discuss the flow of control and data during I/O exit processing by focusing on the call type as determined by the Function Code.

Information that can be returned from the exit

The I/O exit should set any non-zero exit return code as appropriate. The Exit Return Code field is initialized to zero on every call from IBM File Export to the exit, so there is no need for you to set it to zero before each call. However, good programming practice suggests setting it to zero anyway.

The I/O exit can also set an informational, warning, or error message into the Message field of the I/O exit control block. The I/O exit can set the Warning Code at any time to tell IBM File Export to continue processing and not call back the I/O exit (Warning Code = 2). The I/O exit will nonetheless be called again, once at Termination and, if an Open was performed, once to call Close. In order for IBM File Export to check the warning code, the I/O exit must set the Exit Return Code to a warning (Exit Return Code = 4).

Exit functions

This section describes the functions performed by the I/O exit.

The Initialization call

For any particular data set, the I/O exit will always be called first with one (and only one) initialization call. This is an opportunity for the exit to indicate the type of processing it will provide for the data set, to allocate storage, and so forth. At initialization, the exit should:

- Decide whether to process the data set. The data set name and other information is available in the I/O Exit control block. If the exit "knows" from the data set name and other information that it should not process the data set because it is not compressed or encrypted, the exit should indicate that IBM File Export should not call the exit any longer.
 - Only at Initialization the Level Support field can be set to 0. This indicates that IBM File Export should not call the I/O exit.

- In response to any I/O exit call, you can set the Exit Return Code to a warning (Exit Return Code=4) and set the Warning Code field =2 to avoid further calls. IBM File Export will then handle the remaining I/O without the use of the I/O exit.

If doing compression or encryption, set the Level Support field to 1. This indicates to IBM File Export that it is to perform the real I/O. Setting the Level Support field equal to 0 indicates that the data is not compressed or encrypted, and IBM File Export will not use the I/O exit at all.

If it is needed, obtain dynamic storage:

- Typically, only an Assembler exit will need to obtain storage. COBOL and PL/I storage and addressability are preserved between calls to the I/O exit.
- IBM File Export provides both a current input and a current output record buffer, so even an Assembler exit may not need to allocate storage.
- Pointers to allocated storage and other information can be saved in the User Scratchpad area at the end of the I/O control block, and will be maintained between Initialization and Termination calls.

There is no reason to change the values in the "Functions supported by Exit" field. Both Read and Write support are required and this is the default.

The Termination call

The last call for a given data set will always be one (and only one) Termination call. The Termination call is intended primarily to allow deallocation of resources allocated at Initialization. IBM File Export will make the Termination call even if exit processing was terminated by an error or a request from the exit to continue without calling the exit again.

The Open call

The Open call is performed only when processing sequential or VSAM data sets. Processing an IMS database will not cause an Open call to be performed. The Open Function Code means the exit is being called immediately after a "real" file open. Additional data set information is now available. The exit should decide whether to continue processing the data set.

The Close call

The Close call is performed only when processing sequential or VSAM data sets. Processing an IMS database will not cause a Close call to be performed. The Close Function Code means the exit is being called immediately prior to a "real" file close.

Read and Write calls

During a Read operation:

IBM File Export reads a compressed or encoded record from DASD and places its address in the I/O exit's Current Input Record field. IBM File Export then calls the I/O exit with a Read function code. The I/O exit should take the current input record and decompress or decode it into the current output record buffer. Upon return from the I/O exit, IBM File Export will use the decompressed or decoded current output record from the I/O exit.

During a Write operation:

IBM File Export takes an in-memory, uncompressed or uncoded record and places its address into the I/O exit Current Input Record field. IBM File Export then calls the I/O exit with a Write function code. The I/O exit should take the current input record and compress or encode it into the current output record buffer. Upon return from the I/O exit, IBM File Export will write the current output record from the I/O exit to DASD.

In addition to setting the current output record data, the I/O exit should also set the current output record length during a Read or Write Operation. If the current output record length is not set, or is set explicitly to zero, any changes in the current output record data buffer will be ignored and the current input record will be used as the current output record.

Appendix G. Troubleshooting

This section contains troubleshooting tips for working with IBM File Export.

LOBs

LOBs are not supported by IBM File Export. Thus, if there are LOBs in the table on which you are evaluating a SELECT statement, the SELECT will fail.

Output record formatting when using the COBOL parser

For information on formatting output records when using the COBOL parser, refer to “How use of the Cobol parser affects output alignment” on page 264.

Appendix H. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10594-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this publication to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is as your own risk.

Licensees of this program who want to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue

San Jose, CA 95141-1003
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

Advanced Peer-to-Peer	IBM
Networking	Redbooks
CICS/ESA	OS/390
DB2	z/OS
DB2 Universal Database	

Other company, product, and service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

Glossary

The following cross-references are used in this glossary:

Contrast with. This refers to a term that has an opposed or substantively different meaning.

Synonym for. This indicates that the term has the same meaning as a preferred term, which is defined in its proper place in the dictionary.

Synonym with. This is a backward reference from a defined term to all other terms that have the same meaning.

See. This refers the reader to multiple-word terms that have the same last word.

See also. This refers the reader to terms that have a related, but not synonymous, meaning.

A

abend. Abnormal end of task.

abend reason code. A 4-byte hexadecimal code that uniquely identifies a problem with DB2. A complete list of DB2 abend codes and their explanations is contained in *DB2 Messages and Codes*.

abnormal end of task (abend). Termination of a task, job, or subsystem because of an error condition that recovery facilities cannot resolve during execution.

active log. The portion of the DB2 log to which log records are written as they are generated. The active log always contains the most recent log records, whereas the archive log holds those records that are older and no longer fit on the active log.

archive log. The portion of the DB2 log that contains log records that have been copied from the active log.

B

backward log recovery. The fourth and final phase of restart processing during which DB2 scans the log in a backward direction to apply UNDO log records for all aborted changes.

bind. The process by which the output from the DB2 precompiler is converted to a usable control structure (which is called a package or an application plan). During the process, access paths to the data are selected and some authorization checking is performed.

bootstrap data set (BSDS). A VSAM data set that contains name and status information for DB2 as well as RBA range specifications, for all active and archive log data sets. It also contains passwords for the DB2 directory and catalog, and lists of conditional restart and checkpoint records.

buffer pool. Main storage that is reserved to satisfy the buffering requirements for one or more table spaces or indexes.

C

catalog. In DB2, a collection of tables that contains descriptions of objects such as tables, views, and indexes.

catalog table. Any table in the DB2 catalog.

clustering index. An index that determines how rows are physically ordered in a table space.

column. The vertical component of a table. A column has a name and a particular data type (for example, CHARACTER, DECIMAL, or INTEGER).

command. A DB2 operator command or a DSN subcommand. A command is distinct from an SQL statement.

customization. The process of describing optional changes to defaults of a software program that is already installed on the system and configured so that it can be used. Contrast with Configuration.

customize. To describe the system, the devices, programs, users, and user defaults for a particular data processing system or network. Contrast with Configure.

D

DASD. Direct access storage device.

data definition name. The name of a data definition (DD) statement that corresponds to a data control block containing the same name.

DB2 catalog. Tables that are maintained by DB2 and that contain descriptions of DB2 objects, such as tables, views, and indexes.

direct access storage device. A device in which access time is independent of the location of the data.

DSN. (1) The default DB2 subsystem name. (2) The name of the TSO command processor of DB2. (3) The first three characters of the DB2 module and macro names.

I

image copy. An exact reproduction of all or part of a table space. DB2 provides utility programs to make full image copies (to copy the entire table space) or incremental image copies (to copy only those pages that have been modified since the last image copy).

index. A set of pointers in DB2 that are logically ordered by the values of a key. Indexes can provide faster access to data and can enforce uniqueness on the rows in a table.

index space. A DB2 page set that is used to store the entries of one index.

ISPF. Interactive System Productivity Facility.

J

JCL. Job control language. Synonym with job control language.

job control language. JCL. A control language that is used to identify a job to an operating system and to describe the job's requirements. Synonym with JCL.

job profile. Customized settings that describe the jobs you want to run.

L

lock. A means of controlling concurrent events or access to data. DB2 locking is performed by the IRLM (Inter-Region Lock Manager).

lock duration. The interval over which a DB2 lock is held.

lock escalation. The promotion of a lock from a row, page, or LOB lock to a table space lock because the number of page locks that are concurrently held on a given resource exceeds a present limit.

locking. The process by which the integrity of DB2 data is ensured. Locking prevents concurrent users from accessing inconsistent data.

log. A collection of records that describe the events that occur during DB2 execution and that indicate their sequence. The information thus recorded is used for recovery in the event of a failure during DB2 execution.

logical unit of work (LUW). The processing that a program performs between synchronization points.

logical unit of work identifier (LUWID). A name that uniquely identifies a thread within a network. This name consists of a fully-qualified LU network name, an LUW instance number, and an LUW sequence number.

log initialization. The first phase of restart processing during which DB2 attempts to locate the current end of the log.

log record sequence number (LRSN). A number that DB2 generates and associates with each log record. DB2 also uses the LRSN for page versioning. The LRSNs that a particular DB2 data sharing group generates form a strictly increasing sequence for each DB2 log and a strictly increasing sequence for each page across the DB2 group.

P

page. A unit of storage within a DB2 table space (4 KB, 8 KB, 16 KB, or 32 KB) or an index space (4 KB). In a table space, a page contains one or more rows of a table. In a LOB table space, a LOB value can span more than one page, but no more than one LOB value is stored on a page.

S

Source. A source is a file (either DB2 table, IMS database, VSAM file, or sequential file) from which data is extracted for the purpose of creating test data.

T

Target. A target is a file that is created using IBM File Export, and is based on the data that is extracted from the source file and your specification of the target output.

Index

Special characters

262

A

ABX\$CLST 119, 164, 197
ABX\$DB2W 107
ABX\$DBCL 116, 117
ABX\$DBCS 118
ABX\$DBJC 129
ABX\$DBQE 114
ABX\$DBQL 113
ABX\$DBQT 125
ABX\$DBRI 95
ABX\$DBST 108
ABX\$DBTC 120
ABX\$DBTL 115
ABX\$DDE1 127, 167, 177
ABX\$DDE2 181
ABX\$DDE4 180
ABX\$DDSL 171
ABX\$DDWZ 128, 168, 178
ABX\$DOPT 106
ABX\$DSLS 141, 142, 144, 145, 147
ABX\$DSSL 130, 200
ABX\$DTC1 121
ABX\$DTC2 121
ABX\$FFDB 141, 145
ABX\$FLST 164, 197
ABX\$FOPT 175
ABX\$FPSR 142
ABX\$FPTG 143, 146, 148
ABX\$FSEL 155
ABX\$FSEQ 188
ABX\$GOPT 30
ABX\$IMSD 166
ABX\$IMSJ 170
ABX\$IMSW 134, 135
ABX\$IOPT 137
ABX\$IPFI 166, 195, 198
ABX\$JCLG 198
ABX\$MAIN 27
ABX\$MEML 130, 199
ABX\$NFLD 157, 189
ABX\$QCRT 124
ABX\$RELC 111, 152, 186
ABX\$RELF 110, 151, 184
ABX\$RELS 109, 150, 183
ABX\$RMNG 28, 35
ABX\$RRRED 122, 165
ABX\$RRLS 123
ABX\$SELE 162, 195
ABX\$SELS 161, 194
ABX\$SEQW 174
ABX\$SLST 163, 196
ABX\$SMNG 29, 102
ABX\$SQLT 160, 193
ABX\$SQTL 159, 192
ABX\$VARS 149, 182
ABX\$VLST 148, 182
ABX\$VSDB 179
ABX\$XDXS 169
ABX\$ZCGF 60, 72, 86
ABX\$ZCNL 52
ABX\$ZCRR 36
ABX\$ZCSF 58, 70, 84
ABX\$ZEGF 56, 68, 82

ABX\$ZERL 93
ABX\$ZESF 54, 66, 80
ABX\$ZFDD 64, 78
ABX\$ZFDL 53, 65, 79
ABX\$ZFRE 94
ABX\$ZFRL 74, 88
ABX\$ZMRL 75, 89
ABX\$ZPRS 97
ABX\$ZRDL 50
ABX\$ZRDS 77, 91
ABX\$ZREC 51
ABX\$ZREE 94
ABX\$ZREL 92
ABX\$ZRNM 37
ABX\$ZSCR 76, 90
ABX\$ZSE1 41
ABX\$ZSE2 42
ABX\$ZSE3 44, 45
ABX\$ZSE4 51
ABX\$ZSE5 43
ABX\$ZSGS 49
ABX\$ZSLS 48, 74
ABX\$ZSTL 38
ABX\$ZSTR 40
ABX\$ZTBS 47
ABX\$ZTLS 46, 62
ABX\$ZTRL 62
ABXCNTLI 267
ABXCNTLO 267
ABXFELD0
 control statements 260
 EXCLDBD control statement, used with 260
 INCLDBD control statement, used with 261
 JCL requirements 260
 OPTIONS control statement, used with 260
 requirements and restrictions 259
 using to insert data to IMS databases 259
ABXSQL 269
ABXUTIL 4
accessibility features x
AGN statement 207
alignment, output 264
 when using COBOL 264
ALTID statement 207
ALTRRC00 statement 207
application-defined relationships 5
 definition of 6
audience for this book ix
AUDIT statement 207

B

BKO statement 208
BUF statement 208

C

CLIST AUZ
 listing 11
COBOL
 aligning output when using 264
 output record formatting with 264
coding examples 281
COLLECT statement 208
Column List panel 52
COLUMN statement 209
Columns Selection Panel 118

- comments
 - non-printable 257
 - printable 257
- comments, placing in JCL 257
- comments, submitting xi
- CONTENT statement 234
- control statement syntax
 - additional information 257
 - LIMIT 258
 - MAXIMUM 262
 - MINIMUM 262
 - Occurs Depending On 262
 - PRINT 262
 - SKIP 258
 - SPUFI 262
 - AGN statement 207
 - ALTID statement 207
 - ALTRRC00 statement 207
 - AUDIT statement 207
 - BKO statement 208
 - BUF statement 208
 - COLLECT statement 208
 - COLUMN statement 209
 - CPUTIME statement 210
 - DATE statement 210
 - DBDLIB statement 212
 - DBRC statement 212
 - DFSVSAMP statement 213
 - DIRCA statement 213
 - DLTDUPS statement 213
 - DOPTLIB statement 214
 - DPSBMAC statement 214
 - DPSBPFX statement 214
 - DPSBSFX statement 215
 - DYNPSB statement 215
 - EXCLDBD, used with ABXFELD0 260
 - EXCLUDE statement 215
 - EXCPVR statement 216
 - EXITACTIVE statement 216
 - EXITNAME statement 216
 - FIELD statement 217
 - FMTO statement 219
 - GSGNAME statement 220
 - IMSID statement 220
 - IMSPLEX statement 220
 - INCLDBD, used with ABXFELD0 261
 - INPCB# statement 221
 - INSERT statement 221
 - IRLM statement 221
 - IRLMNM statement 222
 - LAST_SEGMENT statement 222
 - LIMIT statement 222
 - LITERAL statement 222
 - LOCKMAX statement 224
 - MON statement 224
 - NBA statement 224
 - OBA statement 225
 - OFIELD statement 225
 - OPT statement 228
 - OPTIONS statement 228
 - OPTIONS, used with ABXFELD0 260
 - OUTPCB# statement 232
 - PARDLI statement 232
 - placing comments in your JCL 257
 - PROCOPT statement 232
 - PSBNAME statement 233
 - RELATIONSHIP statement 233
 - RESLIB1 statement 236
 - RESLIB2 statement 236
 - RESLIB3 statement 236
 - RESULTSET 236
 - SEGMENT statement 237
 - SELECT statements 239
 - SKIP statement 244
 - SOURCEDB2 statement 244

- SOURCEDBD statement 244, 248
- SOURCEDD statement 245
- SSM statement 245, 256
- STIMER statement 245
- SWAP statement 246
- TABLE statement 246
- TARGETDB2 statement 248
- THEN INSERT statement 248
- TYPE statement 251
- VARIABLE statement 254
- control statements
 - hyphen within a name 204
 - order of statements 203
- control syntax
 - creating without using ISPF 203
 - definitions and descriptions 206
 - INSERT statement
 - requirements associated with 251
 - slack bytes 261
 - JCL comprising 203
 - list of statements 204
 - ordering of statements in 203
 - RELATIONSHIP statement
 - CONTENT 234
- conventions used in this book ix
- CPUTIME statement 210
- Create Group Field panel 60, 72, 86
- Create Record panel 51
- Create Repository panel 36
- Create Simple Field panel 58, 70, 84
- Create Storage panel 40
- criteria, selection
 - specifying 103

D

- data manipulation engine 4
- Data Set Allocation Parameters panel 128, 168, 178
- data, exporting 101
- DATE statement 210
- DB2
 - control syntax used with 204
 - DB2 data, exporting 105
 - DB2 Menu panel 107
 - DB2 Product Options panel 106
 - DB2 RI definitions, specifying 94
 - DB2 Source and Target panel 108
 - DB2 Source Data Functions panel 107
 - DB2 storage
 - create or edit 40
 - DB2 table definitions, specifying 94
- DBDLIB statement 212
- DBRC statement 212
- DD Statement Definition panel 127, 167, 177, 180, 181
- DD Statements List panel 166
- DDnames used by File Export 267
 - ABXCNTLI 267
 - ABXCNTLO 267
 - ABXIN 267
 - ABXOUT 268
 - ABXSQL 269
 - IMSRPT 269
 - NEWIMS 269
 - OLDIMS 269
 - STEBLIB 270
 - SYSABEND 270
 - SYSPRINT 268
 - SYSUDUMP 270
 - TABLnnn 270
- ddnames used by File Export 4
- define field manipulation
 - sequential 187
 - VSAM 187
- define field specification

- sequential 187
- VSAM 187
- define relationships
 - IMS full function 150
 - sequential 183
 - VSAM 183
- define segment manipulation
 - IMS full function 154, 170
- define segment specification
 - IMS full function 154, 170
- define type and SELECT statement manipulation
 - sequential 191
 - VSAM 191
- define type and SELECT statement specification
 - sequential 191
 - VSAM 191
- define variables and literals
 - sequential 181
 - VSAM 181
- DFSVSAMP statement 213
- DIRCA statement 213
- DL/I
 - control syntax used with 204
- DLTDUPS statement 213
- DOPTLIB statement 214
- DPSBMAC statement 214
- DPSBPFX statement 214
- DPSBSFX statement 215
- DYNPSB statement 215

E

- Edit DB2 Storage panel 41
- Edit Group Field panel 56, 68, 82
- Edit IMS Storage panel 42, 43
- Edit Query Criteria panel 124
- Edit Query panel 114
- Edit Record panel 51
- Edit Relation Fields panel 94
- Edit Relation panel 93, 122, 165
- Edit Select Statement panel 162, 195
- Edit Simple Field panel 54, 66, 80
- Edit Target Column Panel 121
- Edit VSAM or Sequential Storage panel 44, 45
- examples, coding 281
- EXCLDBD statement
 - used with ABXFELD0 260
- EXCLUDE statement 215
- EXCPVR statement 216
- EXITACTIVE statement 216
- EXITNAME statement 216
- exporting data using IBM File Export for z/OS 101
- exporting DB2 data 105
- exporting IMS data 133
- exporting sequential data 173
- exporting VSAM data 173
- external DDnames required by File Export
 - DD statements from FABCUR6 270
 - ACBLIB 271
 - DURDzzzE 271
 - DURDzzzO 271
 - DURSzzzO 271
 - IMSACBA 271
 - IMSACBB 271
 - MODSTAT 271
 - MODSTAT2 271
 - RMODLIB 271
 - UR6AUDIT 271
 - UR6CTL 271
 - UR6DBDFN 271
 - UR6PRINT 271
 - XDzzzzzE 271
 - XDzzzzzO 271
 - XSzzzzzO 271

- DD statements from FABCUR7 271
 - RMODLIB 272
 - UR7AUDIT 272
 - UR7CTL 272
 - UR7DATA 272
 - UR7DBDFN 272
 - UR7PRINT 272

F

- Fast Path
 - control syntax used with 204
- Field List panel 53, 64, 65, 78, 79
- field manipulation
 - sequential 187
 - VSAM 187
- field specification
 - sequential 187
 - VSAM 187
- Field Specification panel 157, 189
- FIELD statement 217
- Fields panel 188
- File Export
 - data manipulation engine 4
 - ddnames used by 4
 - design objectives 3
 - functions of 3
 - hardware requirements 9
 - interfaces to 4
 - referential integrity in 5
 - software requirements 9
- Filter Specification panel 110, 151, 184
- FMTO statement 219
- formatting, output record 263
 - when using COBOL 264
- full function, IMS
 - define relationships 150
 - define segment specification and manipulation 154
 - JCL generation 170

G

- generation, JCL
 - IMS full function 170
 - sequential 198
 - VSAM 198
- getting started with IBM File Export for z/OS 101
- GSGNAME statement 220

H

- hardware requirements 9

I

- IBM File Export for z/OS control syntax 203
- Import DB2 Definitions panel 95
- IMS
 - control syntax used with 204
- IMS data, exporting 133, 173
- IMS direct processing
 - ABXFELD0 259
- IMS full function
 - define relationships 150
 - define segment specification and manipulation 154
 - JCL generation 170
 - segment specification and manipulation 154
 - Target Database Specification panel 145
- IMS Menu panel 134, 135
- IMS Session Settings panel 137
- IMS Source Database Specification panel 141, 142
- IMS storage

- create or edit 41
- IMS Target Database Specification panel 143, 145, 146, 148
- IMSID statement 220
- IMSPLEX statement 220
- IMSRTPT 269
- INCLDBD statement
 - used with ABXFELD0 261
- INPCB# statement 221
- INSERT statement 221
 - requirements associated with 251
 - slack bytes 261
- Installation Verification Programs 273
 - DB2 273
 - IMS 273
 - sequential files 274
 - VSAM 274
- interfaces into File Export 4
- IRLM statement 221
- IRLMNM statement 222
- IVPs 273
 - DB2 273
 - IMS 273
 - sequential files 274
 - VSAM 274

J

- JCL generation
 - IMS full function 170
 - sequential 198
 - VSAM 198
- JCL Generation for IMS Session panel 170
- JCL Generation panel 129, 198
- JCL, control syntax 203

L

- LAST_SEGMENT statement 222
- LIMIT
 - additional information 258
- LIMIT statement 222
- limiting output 258
- listing
 - CLIST AUZ 11
- LITERAL statement 222
- literals and variables, define
 - sequential 181
 - VSAM 181
- LOBs 301
- LOCKMAX statement 224
- LookAt message retrieval tool x

M

- Main Menu panel 27
- Manage Record Layouts panel 75, 89
- manipulation, field
 - sequential 187
 - VSAM 187
- manipulation, segment
 - IMS full function 154
- manipulation, type and SELECT statement
 - sequential 191
 - VSAM 191
- MAXIMUM
 - additional information about 262
- Member Selection List panel 130, 199
- message retrieval tool, LookAt x
- MINIMUM
 - additional information about 262
- MON statement 224

N

- NBA statement 224
- New Field Selection panel 164
- NEWIMS 269
- Notices 303

O

- OBA statement 225
- Occurs Depending On 262
- OFIELD statement 225
- OLDIMS 269
- OPT statement 228
- OPTIONS statement 228
 - used with ABXFELD0 260
- OUTPCB# statement 232
- output alignment 264
 - when using COBOL 264
- output record formatting 263
 - when using COBOL 264
- output, limiting 258

P

- panel
 - ABX\$CLST 119, 164, 197
 - ABX\$DB2W 107
 - ABX\$DBCL 116, 117
 - ABX\$DBCS 118
 - ABX\$DBJC 129
 - ABX\$DBQE 114
 - ABX\$DBQL 113
 - ABX\$DBQT 125
 - ABX\$DBRI 95
 - ABX\$DBTC 120
 - ABX\$DBTL 115
 - ABX\$DDE1 127, 167, 177
 - ABX\$DDE2 181
 - ABX\$DDE4 180
 - ABX\$DDWZ 128, 168, 178
 - ABX\$DOPT 106
 - ABX\$DSBG 107
 - ABX\$DSLS 141, 142, 144, 145, 147
 - ABX\$DSSL 130, 171, 200
 - ABX\$DTC1 121
 - ABX\$DTC2 121
 - ABX\$FFDB 141, 145
 - ABX\$FLST 164, 197
 - ABX\$FOPT 175
 - ABX\$FPSR 142
 - ABX\$FPTG 143, 146, 148
 - ABX\$FSEL 155
 - ABX\$FSEQ 188
 - ABX\$GOPT 30
 - ABX\$IMSD 166
 - ABX\$IMSJ 170
 - ABX\$IMSW 134, 135
 - ABX\$IOPT 137
 - ABX\$IPFI 166, 195, 198
 - ABX\$JCLG 198
 - ABX\$MAIN 27
 - ABX\$MEML 130, 199
 - ABX\$NFLD 157, 189
 - ABX\$QCRT 124
 - ABX\$RELC 111, 152, 186
 - ABX\$RELF 110, 151, 184
 - ABX\$RELS 109, 150, 183
 - ABX\$RMNG 28, 35
 - ABX\$RRED 122, 165
 - ABX\$RRLS 123
 - ABX\$SELS 161, 162, 194, 195
 - ABX\$SEQW 174

ABX\$SLST 163, 196
 ABX\$SMNG 29, 102
 ABX\$SQLT 160, 193
 ABX\$SQT 159, 192
 ABX\$VARS 149, 182
 ABX\$VLST 148, 182
 ABX\$VSDB 179
 ABX\$DXS 169
 ABX\$ZCGF 60, 72, 86
 ABX\$ZCNL 52
 ABX\$ZCRR 36
 ABX\$ZCSF 58, 70, 84
 ABX\$ZEGF 56, 68, 82
 ABX\$ZERL 93
 ABX\$ZESF 54, 66, 80
 ABX\$ZFDD 64, 78
 ABX\$ZFDL 53, 65, 79
 ABX\$ZFRE 94
 ABX\$ZFRL 74, 88
 ABX\$ZMRL 75, 89
 ABX\$ZPRS 97
 ABX\$ZRDL 50
 ABX\$ZRDS 77, 91
 ABX\$ZREC 51
 ABX\$ZREE 94
 ABX\$ZREL 92
 ABX\$ZRNM 37
 ABX\$ZSCR 76, 90
 ABX\$ZSE1 41
 ABX\$ZSE2 42
 ABX\$ZSE3 44, 45
 ABX\$ZSE4 51
 ABX\$ZSE5 43
 ABX\$ZSGS 49
 ABX\$ZSLS 48
 ABX\$ZSTL 38
 ABX\$ZSTR 40
 ABX\$ZTBS 47
 ABX\$ZTLS 46, 62, 74
 ABX\$ZTRL 62
 Columns Selection 118
 Data Set Allocation Parameters 128, 168, 178
 Edit Query 114
 Edit Select Statement 162, 195
 Edit Target Column 121
 Member Selection List 130, 199
 New Field Selection 164
 Queries 113
 Select Statement List 161, 194
 Source Table 115, 116, 117
 SQL Output Parameters 166, 195, 198
 Target Columns 120
 Target Specification 125
 panel, Column List 52
 panel, Create Group Field 60, 72, 86
 panel, Create Record 51
 panel, Create Repository 36
 panel, Create Simple Field 58, 70, 84
 panel, Create Storage 40
 panel, DB2 Menu 107
 panel, DB2 Product Options 106
 panel, DB2 Source and Target 108
 panel, DB2 Source Data Functions 107
 panel, DD Statement Definition 127, 167, 177, 180, 181
 panel, DD Statements List 166
 panel, Edit DB2 Storage 41
 panel, Edit Group Field 56, 68, 82
 panel, Edit IMS Storage 42, 43
 panel, Edit Query Criteria 124
 panel, Edit Record 51
 panel, Edit Relation 93, 122, 165
 panel, Edit Relation Fields 94
 panel, Edit Simple Field 54, 66, 80
 panel, Edit VSAM or Sequential Storage 44, 45
 panel, Field List 53, 64, 65, 78, 79
 panel, Field Specification 157, 189
 panel, Fields 188
 panel, Filter Specification 110, 151, 184
 panel, Import DB2 Definitions 95
 panel, IMS Menu 134, 135
 panel, IMS Session Settings 137
 panel, IMS Source Database Specification 141, 142
 panel, IMS Target Database Specification 143, 146, 148
 panel, JCL Generation 129, 198
 panel, JCL Generation for IMS Session 170
 panel, Main Menu 27
 panel, Manage Record Layouts 75, 89
 panel, Parser Configuration 97
 panel, Record Layout (DB2) 62
 panel, Record Layout (IMS) 74
 panel, Record Layout (VSAM, Sequential) 88
 panel, Record Layouts 50
 panel, Record Layouts Selection 77, 91
 panel, Record Relations 92
 panel, Record Type 193
 panel, Record Type List 159, 192
 panel, Relation Content 111, 152, 186
 panel, Relation Field List 94
 panel, Repository Management 28, 35
 panel, Repository Selection 37
 panel, Segment List 48, 74
 panel, Segments Selection 49
 panel, Selection Criteria 76, 90, 119, 164, 197
 panel, Sequential & VSAM Menu 174
 panel, Sequential Data Sets Specification 179
 panel, Session Management 29, 102
 panel, Session Settings
 Sequential and VSAM 175
 panel, Settings 30
 panel, Sort Control Statements 169
 panel, SQL Type 160
 panel, Statement List 123
 panel, Storage List 38
 panel, Table List 46, 62
 panel, Tables Selection 47
 panel, Variables and Literals 149, 182
 panels
 ABX\$DBST 108
 PARDLI statement 232
 parser
 used with repository 96
 Parser Configuration panel 97
 PROCOPT statement 232
 PSBNAME statement 233

Q

Queries panel 113

R

record formatting, output 263
 Record Layout (DB2) panel 62
 Record Layouts (IMS) panel 74
 Record Layouts (VSAM/Sequential) panel 88
 Record Layouts panel 50
 Record Layouts Selection panel 77, 91
 Record Relations panel 92
 record relationships
 managing in the repository 92
 Record Type List panel 159, 192
 Record Type panel 193
 referential integrity
 application-defined relationships 6
 defined by DB2 catalog 5
 parent-child relationships in DB2 5
 Referential integrity in IBM File Export for z/OS 5
 Relation Content panel 111, 152, 186
 Relation Field List panel 94

- RELATIONSHIP statement 233
 - CONTENT 234
- relationships, application-defined 5
 - definition of 6
- relationships, define
 - IMS full function 150
 - sequential 183
 - VSAM 183
- repository
 - creating 36
 - DB2 storage
 - create 40
 - edit 40
 - DB2 storage structure
 - opening 45
 - working with 45
 - deleting 37
 - IMS storage
 - create 41
 - edit 41
 - IMS storage structure
 - opening 47
 - working with 47
 - managing record relationships in 92
 - opening 37
 - parser used with 96
 - record layouts
 - DB2 52
 - defining 39
 - IMS 63
 - sequential 78
 - VSAM 78
 - working with 49
 - sequential storage
 - create 44
 - edit 44
 - storage structure
 - adding 39
 - creating 38
 - deleting 39
 - editing 39
 - opening 39
 - refreshing 39
 - VSAM storage
 - create 44
 - edit 44
 - VSAM storage structure
 - working with 49
 - working with 35
- Repository Management panel 28, 35
- Repository Selection panel 37
- repository, File Export 5
- requirements
 - hardware 9
 - software 9
- RESLIB1 statement 236
- RESLIB2 statement 236
- RESLIB3 statement 236
- RESULTSET statement 236
- RI definitions, specifying 94
- RI in IBM File Export for z/OS 5

S

- scrambling, about 257
- screen readers and magnifiers x
- Segment List panel 48, 74
- segment manipulation
 - IMS full function 154
- segment specification
 - IMS full function 154
- SEGMENT statement 237
- Segments Selection panel 49
- SELECT NOTHING control statement 243
- select session, IMS 135
- Select Statement List panel 161, 194
- SELECT statements 239
- Selection Criteria panel 76, 90, 119, 164, 197
- selection criteria, specifying 103
- sequential
 - define field specification and manipulation 187
 - define relationships 183
 - define type and SELECT statement specification and manipulation 191
 - define variables and literals 181
 - JCL generation 198
 - select session 106
 - specify source data set 178
 - specify target data set 178
 - summary 173
- Sequential & VSAM Menu panel 174
- sequential data
 - control syntax used with 204
- Sequential Data Sets Specification panel 179
- sequential storage
 - create or edit 44
- Session Management panel 29, 102
- Session Settings panel
 - Sequential and VSAM 175
- session, select
 - IMS 135
- session, select type of 106
- Settings panel 30
- SKIP
 - additional information 258
- SKIP statement 244
- slack bytes 261
- software requirements 9
- Sort Control Statements panel 169
- Source Table Panel 115, 116, 117
- source, definition of 3
- SOURCEDB2 statement 244
- SOURCEDBD statement 244, 248
- SOURCEDD statement 245
- sources and targets, valid 101
- specification, field
 - sequential 187
 - VSAM 187
- specification, segment
 - IMS full function 154
- specification, type and SELECT statement
 - sequential 191
 - VSAM 191
- specify source data set
 - sequential 178
 - VSAM 178
- specify target data set
 - VSAM 178
- SPUFI
 - additional information about 262
- SQL Output Parameters Panel 166, 195, 198
- SQL Type panel 160
- SSM statement 245, 256
- Statement List panel 123
- statements, control
 - hyphen within a name 204
 - order of statements 203
- STEPLIB DD 270
- STIMER statement 245
- Storage List panel 38
- summary
 - sequential 173
 - VSAM 173
- SWAP statement 246
- SYSABEND 270
- SYSPRINT 268
- SYSUDUMP 270

T

- Table List panel 46, 62
- TABLE statement 246
- Tables Selection panel 47
- TABLnnn 270
- Target Columns Panel 120
- target data set
 - sequential 178
 - VSAM 178
- Target Specification panel 125
- target, definition of 3
- TARGETDB2 statement 248
- targets and sources, valid 101
- tasks supported by this book ix
- terminology X
- THEN INSERT statement 248
- troubleshooting
 - LOBs 301
 - output record formatting when using COBOL 301
- type and SELECT statement manipulation
 - sequential 191
 - VSAM 191
- type and SELECT statement specification
 - sequential 191
 - VSAM 191
- TYPE statement 251

V

- valid sources and targets 101
- VARIABLE statement 254
- Variables and Literals panel 149, 182
- variables and literals, define
 - sequential 181
- VSAM
 - define field specification and manipulation 187
 - define relationships 183
 - define type and SELECT statement specification and manipulation 191
 - define variables and literals 181
 - JCL generation 198
 - select session 106
 - specify source data set 178
 - specify target data set 178
 - summary 173
- VSAM files
 - control syntax used with 204
- VSAM storage
 - create or edit 44

Readers' Comments—We'd Like to Hear from You

IBM File Export for z/OS
Reference Manual
Version 1 Release 1

Publication No. SC31-6870-02

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

Thank you for your response. May we contact you? Yes No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

_____ Name	_____ Address
_____ Company or Organization	_____
_____ Phone No.	_____



Cut or Fold
Along Line

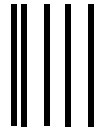
SC31-6870-02

International Business Machines Corporation
IBM Corporation --Silicon Valley Laboratory
Department DTX/E269
555 Bailey Avenue
San Jose, CA
U.S.A.
95141 - 9989

Fold and Tape

Please do not staple

Fold and Tape



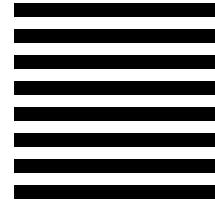
NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

SC31-6870-02
INTERNATIONAL BUSINESS MACHINES CORP
Silicon Valley Laboratory -- Department DTX/E269
555 Bailey Avenue
San Jose, CA 95141-9989



Fold and Tape

Please do not staple

Fold and Tape

Cut or Fold
Along Line



Program Number: 5697-112

SC31-6870-02

