

FILEXTRACT
ADMINISTRATOR'S GUIDE



Attachmate®

DATABridge™



July 2010

© 2010 Attachmate Corporation. All rights reserved.

No part of the documentation materials accompanying this Attachmate software product may be reproduced, transmitted, transcribed, or translated into any language, in any form by any means, without the written permission of Attachmate Corporation. The content of this document is protected under copyright law even if it is not distributed with software that includes an end user license agreement. The content of this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Attachmate Corporation. Attachmate Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this document.

Attachmate and the Attachmate logo are registered trademarks and DATABridge is a trademark of Attachmate Corporation in the USA.

All other trademarks, trade names, or company names referenced herein are used for identification only and are the property of their respective owners.

A copy of the Attachmate software license agreement governing this product can be found in a 'license' file in the root directory of the software distribution.

Third party notices (if any) can be found in a 'thirdpartynotices' file in the root directory of the software distribution.

Attachmate Corporation
1500 Dexter Avenue North
Seattle, WA 98109 USA
+1.206.217.7100
<http://www.attachmate.com>

Contents

	About This Guide	vii
	Audience	viii
	Conventions	ix
	Abbreviations	x
	Related Documentation	xi
Chapter 1	Introducing FileXtract	1
	Introducing FileXtract	2
	Advantages of FileXtract	3
	Understanding the DATABridge Host Components	5
	Getting Started with FileXtract	6
Chapter 2	Installing FileXtract	9
	Required Hardware and Software	10
	Before You Install	11
	Installing FileXtract	14
	Step 1: Determine the Usercode	14
	Step 2: Determine the Pack	14
	Step 3: Copy the FileXtract Installation WFL	15
	Step 4: Start the FileXtract Installation WFL	15
	Step 5: Establish DBEngine as a System Library (SL)	17
	Installation Results	18

Chapter 3	Using the Sample Reader Libraries	23
	Understanding the Sample Reader Libraries	24
	SUMLOG Reader Library	25
	TTRAIL Reader Library	25
	PRINTFILE Reader Library	26
	BICSS Reader Library	26
	DISKFILE Reader Library	28
	LINCLOG Reader Library	29
	BANKFILE Reader Library	30
	USERDATA Reader Library	30
	TEXT Reader Library	31
	Using the Sample Reader Libraries.	32
 Chapter 4	 Using the COBOL-to-DASDL Utility	 37
	Understanding the COBOL-to-DASDL Utility	38
	BANKFILE COBOL-to-DASDL Example	39
	Generate DB Layout from COBOL 01-Level File Record Desc.	42
	Generating Database Layout from LINC Activity Log Files	46
 Chapter 5	 Creating a Custom Reader Library	 49
	Understanding When to Create a Custom Reader Library	50
	Creating a Custom Reader Library	51
 Appendix A	 Troubleshooting	 53
	General Troubleshooting Procedures	54
	Error Messages	56
 Appendix B	 Out-of-Order Processing	 57
	Out-of-Order Processing	58
	Before Processing.	58
	Processing Archived SUMLOGs.	59
	Processing Current SUMLOGs.	59
	 Glossary	 61
	 Index	 65

About This Guide

This guide contains instructions for installing, configuring, and using Attachmate DATABridge FileXtract (hereafter referred to as FileXtract). This preface includes information to help you use this guide. The following sections are included in this preface:

Audience	vi
Conventions	vii
Related Documentation	ix
Related Documentation	ix

Audience

To install, configure, and run FileXtract, you must be a system administrator, and you must be thoroughly familiar with the following:

- Standard Unisys® operations for MCP-hosted mainframes such as the CS7xxx series, Libra series, ClearPath® NX/LX or A Series
- DMSII databases and Data and Structure Definition Language (DASDL)
- File layouts and the description of those layouts for the files you will be replicating.

Conventions

This guide uses the following conventions:

- Text that you type as well as messages and prompts that appear on the screen are shown in `this type style`.
- In addition to emphasizing text, *italic* indicates variables. For example, if you were asked to type `START WFL/FILEBRIDGE/COBOLTODASDL (“dbname”)`, you would type the actual name of the database in place of the italicized word.
- Actual file names and parameter names are in CAPITAL LETTERS, as follows:
`DATA/GENFORMAT/dbname/CONTROL`
- Optional items in a command are enclosed in [square brackets]. If you include the optional items, do not type the brackets.
- The terms *host* and *mainframe* are used interchangeably to refer to a Unisys MCP-hosted mainframe such as the CS7xxx series, Libra series, ClearPath NX/LX series, or A series



Caution: This caution icon indicates that there is a possibility of losing data or corrupting files. When you see this caution icon, follow the instructions carefully.

Abbreviations

The following abbreviations are used throughout this guide and are provided here for quick reference.

Abbreviation	Name
ABSN	Audit block serial number
AFN	Audit file number
API	Application programming interface
BICSS	Billing interface customer services and statistics
COMS	Communications management system
DASDL	Data and structure definition language
DMSII	Data management system II
INX	Index
LINCLOG	LINC Activity log files
MCP	Master control program
PRINTFILE	Printer backup file
SL	System library
SUMLOG	System summary log
TTRAIL	Transaction trail
WFL	Work flow language

Related Documentation

The following is a list of the documentation you might need to consult when using FileXtract.

FileXtract Readme File

The FileXtract Readme file contains important information, including any information that became available after this guide was written. You can view the FileXtract Readme file in either of the following locations:

Installation DVD Location	Installed Location
Docs\readme_fx.txt	DATA/FILEBRIDGE/README

***DATA*Bridge Host Administrator's Guide**

You can use FileXtract with the *DATA*Bridge Client, DBSpan, or DBSnapshot to replicate flat files. For example, to replicate flat files to a *DATA*Bridge Client, FileXtract uses DBServer, DBEngine, and DBSupport, all of which are *DATA*Bridge host components. For detailed information about *DATA*Bridge host components, refer to the *DATA*Bridge Host Administrator's Guide, which is located in the DOCS folder on your *DATA*Bridge installation DVD.

***DATA*Bridge Clients Administrator's Guide**

You can use the *DATA*Bridge Client with FileXtract to populate a relational database from flat files. *DATA*Bridge Clients are documented in the *DATA*Bridge Client Administrator's Guide, which is located in the DOCS folder on your *DATA*Bridge installation DVD.

***DATA*Bridge Programmer's Reference**

The *DATA*Bridge Programmer's Reference is necessary if you will be creating a custom Reader library. For more information, refer to the *DATA*Bridge Programmer's Reference, which is located on your *DATA*Bridge installation DVD.

Unisys Mainframe

You should have available standard Unisys ClearPath NX/LX series or A Series, DMSII, WFL, and CANDE documentation. This guide lists only general instructions. If you are not completely familiar with DMSII configuration, for example, you may need to refer to the Unisys documentation.

About This Guide

Introducing FileXtract

1

In This Chapter

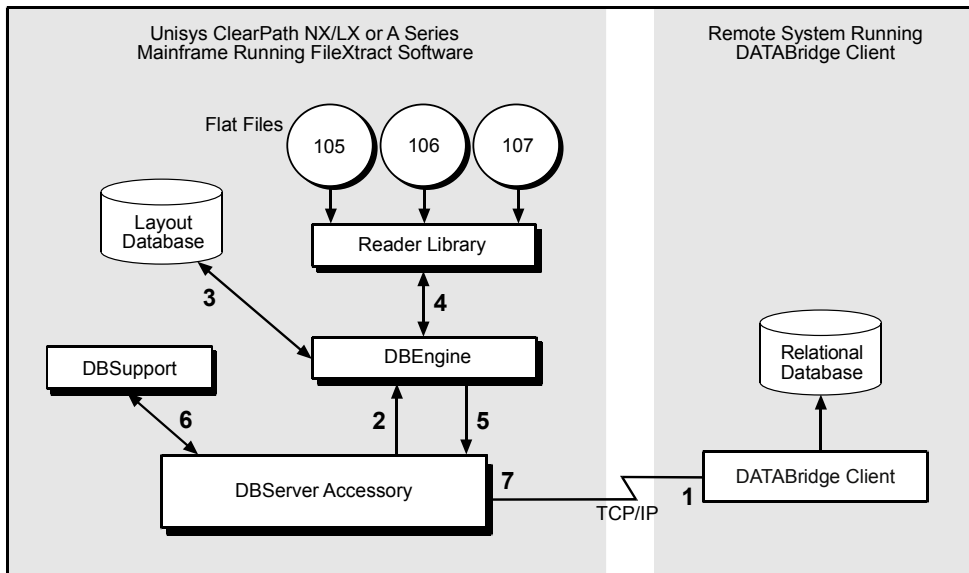
This chapter explains what FileXtract is, how it works, and how to use it. The following sections are included in this chapter:

Introducing FileXtract	2
Getting Started with FileXtract	6

Introducing FileXtract

FileXtract is bundled with the DATABridge host software. FileXtract includes several Reader libraries and other associated files that enable you to replicate (clone and then update) non-DMSII data (hereafter referred to as flat files) residing on Unisys MCP servers.

You can use FileXtract with the DATABridge Client, DBSpan, or DBSnapshot to replicate flat files. The following diagram shows how the DATABridge Client uses FileXtract to replicate flat files.



- 1 The DATABridge Client calls DBServer Accessory to replicate the specified flat files.
- 2 DBServer Accessory calls DBEngine to extract the data for the selected flat files.
- 3 DBEngine accesses the specified FileXtract layout database to determine the layout of the flat files.

No updates are actually done to this database. DBEngine gets all of the layout information it needs from the DESCRIPTION file. The DMSII CONTROL file and the DMSUPPORT library will exist, but the data sets and audit files will not.

- 4 DBEngine calls the appropriate Reader library, as specified in the layout database, to extract the data from the flat files.
- 5 DBEngine sends the data to DBServer Accessory.
- 6 DBServer Accessory calls DBSupport to filter, alter, or transform the data, if needed.
- 7 DBServer Accessory sends the data to the DATABridge Client.

The DATABridge Client populates the relational database and then either waits for additional flat file information or terminates.

Advantages of FileXtract

The advantages of using FileXtract are as follows:

- You can replicate any type of flat file to a Client database.
 - Sample Reader libraries are provided for system summary log files (SUMLOG), COMS Transaction Trail files (TTRAIL), printer backup files (PRINTFILES), BICSS log files (BICSS), LINC Activity logs (LINCLOG), and the USERDATA system file. You can use these sample Reader libraries without any modifications to the Reader libraries, and DASDL files are provided to generate the corresponding layout database. See [Chapter 3, “Using the Sample Reader Libraries,”](#) beginning on page 25 for more information.
 - The DISKFILE Reader library is provided to replicate COBOL-created flat files. If you have COBOL 01-level file record descriptions for your flat files, you can use the COBOL-to-DASDL utility to generate the layout database. See [Chapter 4, “Using the COBOL-to-DASDL Utility,”](#) beginning on page 39 for more information.
 - Sample Reader libraries are also provided that you can customize to replicate any other type of flat file. See [Chapter 5, “Creating a Custom Reader Library,”](#) beginning on page 51 for more information.
- Copying the non-DMSII data offloads decision support, queries, and reporting from the primary host to a database on a remote server, such as a UNIX® or Microsoft® Windows® server.

- The secondary database provides a secure way to make data available to selected individuals, departments, or sites while protecting the flat files on the host.
- The data is available on the Client system even if the host is down or the data communication connection is broken. This eliminates long wait times for data availability. Users can use any database tool available on the Client system to access the data in the secondary database.
- Cloning is required only one time. After the flat files are cloned, FileXtract updates the Client database as new records and flat files become available.

Note: FileXtract cannot track updates to flat file records. If record modifies or deletes occur, you must reclone the flat files.

- You can select what you want to replicate from the flat files; it is not necessary to replicate the entire flat file.

**Understanding the
DATABridge Host
Components**

Each DATABridge Host component in the diagram in the previous section is explained in this section. Refer to the *DATABridge Host Administrator's Guide* for instructions on configuring and using DATABridge components.

DATABridge Engine

The DATABridge Engine (DBEngine) is the main part of the DATABridge host database replication software. DBEngine is a host system library that retrieves structural and layout information from the FileXtract DMSII layout database and initiates the FileXtract Reader libraries to extract the flat file data. DBEngine passes the extracted information to DBServer Accessory, DBSpan, DBSnapshot, or some other DATABridge Accessory.

DBSupport Library

The DBSupport Library provides filtering and translation to the DBServer Accessory (as well as other DATABridge Accessories). After DBServer Accessory receives data from DBEngine, it passes the data to DBSupport for filtering. In brief, DBSupport offers the following to FileXtract:

- Virtual data sets
Virtual data sets appear as normal data sets to DBServer Accessory, DBSpan, and DBSnapshot even though they do not actually exist in the flat files.
- Data set filtering
- Column filtering
- Row filtering

For a complete explanation of data set, column, and row filtering, see DBSupport in the *DATABridge Host Administrator's Guide*.

DBServer Accessory

The DBServer Accessory provides communications between DBEngine on the host and the DATABridge Client. The DATABridge Client requests updates from the DBServer Accessory, which in turn calls DBEngine to retrieve new records. The DATABridge Client then updates the client database with those new records.

DBSpan Accessory

DBSpan produces a replication of one or more data sets into flat sequential disk files. DBSpan updates the cloned flat files by appending the changes to the end of the flat files (unlike DBSnapshot, which replaces the changed records).

DBSnapshot
Accessory

DBSnapshot produces a one-time replication of one or more data sets into flat sequential disk files. DBSnapshot clones the selected data sets each time you run it.

Getting Started with FileXtract

The following is a description of how to use FileXtract to replicate flat files.

- 1 Install FileXtract.
See “[Installing FileXtract](#)” beginning on page 14 for instructions.
- 2 Decide which flat files you want to replicate.
- 3 Determine which Reader library you need to use, then select or generate a layout database for the flat files you want to replicate and start the replication process.

For these types of files	Use this Reader library
System summary log files	SUMLOG See “ Using the Sample Reader Libraries ” beginning on page 34 for instructions.
COMS Transaction Trail files	TTRAIL See “ Using the Sample Reader Libraries ” beginning on page 34 for instructions.
Printer backup files	PRINTFILE See “ Using the Sample Reader Libraries ” beginning on page 34 for instructions.
BICSS log files	BICSS See “ Using the Sample Reader Libraries ” beginning on page 34 for instructions.
Flat file using a COBOL FD	DISKFILE See “ Generating Database Layout from COBOL 01-Level File Record Descriptions ” beginning on page 44 for instructions.

For these types of files	Use this Reader library
LINC Activity logs created by LINC systems	LINCLOG See “Generating Database Layout from LINC Activity Log Files” beginning on page 48 for instructions.
System USERDATA file	USERDATA See “Using the Sample Reader Libraries” beginning on page 34 for instructions.
Flat files that contain information in a “proprietary” format (for example, a flat file that requires reading the <i>n</i> th word to find the address of the next record)	A custom, user-written Reader library See “Creating a Custom Reader Library” beginning on page 53 for instructions.
The BANKFILE sample flat file supplied with FileXtract	BANKFILE See “BANKFILE COBOL-to-DASDL Example” beginning on page 41 for instructions.

- 4 If you will be using DBServer Accessory, define the SOURCE and READER options in the DBServer parameter file. If you will be using DBSpan or DBSnapshot, define the READER option in the Accessory’s parameter file.

Refer to the *DATABridge Host Administrator’s Guide* for instructions.

- 5 Run the Accessory (DBServer, DBSpan, or DBSnapshot).

Refer to the *DATABridge Host Administrator’s Guide* for instructions on configuring and running DATABridge Accessories.

- 6 If you are replicating to the DATABridge Client, run the DATABridge Client DEFINE command against the SOURCE option in DBServer Accessory.

Refer to the *DATABridge Client Administrator’s Guide* for instructions on running the DATABridge Client.

Installing FileXtract

2

In This Chapter

This chapter explains how to install FileXtract. The following sections are included in this chapter:

Required Hardware and Software	10
Before You Install	11
Installing FileXtract	14
Installation Results	18

Required Hardware and Software

FileXtract requires the hardware and software listed below.

Hardware

FileXtract runs on all Unisys MCP-hosted servers.

Software

FileXtract requires the following software:

- MCP level SSR 49.1 or later
- DMSII software 49.1 or later (including the DMALGOL compiler)
- Access to the flat files
- Flat files that adhere to the following criteria:
 - Flat files have the same layout and a common naming convention
 - Flat files consist of only creates added to the end of the file, not modifications to existing records
- Flat files in a group have the same Reader type (interface); for example, all are SUMLOG files

Before You Install

If the DATABridge host database replication software is installed on the mainframe where you want to install FileXtract, it is recommended that you install FileXtract to the same usercode and pack as DATABridge.



Caution: The FileXtract installation WFL overwrites the DATABridge host database replication software files listed in this section; therefore, if you made changes to any of these files, write down the changes before you start the FileXtract installation WFL. When the installation WFL finishes, update the new files to match the settings in the previous ones.

DBSupport

If you changed SYMBOL/DATABRIDGE/SUPPORT, note the changes you made and make the corresponding changes to the new SYMBOL/DATABRIDGE/SUPPORT.

DATABridge Compile WFL

If you changed WFL/DATABRIDGE/COMP, note the changes you made and make the corresponding changes to the new WFL/DATABRIDGE/COMP.

Option	Your Setting
QUEUE =	
STARTTIME =	
BDNAME =	
Other:	

Other DATABridge WFL Files

In all of the WFL/DATABRIDGE files, you can modify the following:

- STARTTIME
- QUEUE
- Name of printer backup files

If you changed any of these items, write your changes in the tables that follow. After installation, update the files accordingly.

WFL/DATABRIDGE/SPAN

Option	Your Setting
QUEUE =	
STARTTIME =	
BDNAME =	

WFL/DATABRIDGE/LISTER

Option	Your Setting
QUEUE =	
STARTTIME =	
BDNAME =	

WFL/DATABRIDGE/SERVER

Option	Your Setting
QUEUE =	
STARTTIME =	
BDNAME =	
USERCODE =	

WFL/DATABRIDGE/SNAPSHOT

Option	Your Setting
QUEUE =	
STARTTIME =	
BDNAME =	

WFL/DATABRIDGE/DBINFO

Option	Your Setting
QUEUE =	
STARTTIME =	
BDNAME =	

SAMPLE Files

If you did not rename SAMPLE files before you changed them, save them with a different name now. The SAMPLE files are as follows:

- DATA/ENGINE/SAMPLE/CONTROL
- DATA/GENFORMAT/SAMPLE/CONTROL
- DATA/LISTER/SAMPLE/CONTROL
- DATA/SERVER/SAMPLE/CONTROL

Note: To use any of the new options, type them in your parameter file.

DBEngine Control File

The DATA/ENGINE/CONTROL file is overwritten with the new DATABridge key. If you made changes to the parameter file, make a note of them now.

Option	Your Setting
KEY or EVALKEY	
Audit Level =	
Property Level =	
Checkpoint client...	
Workers =	
Available From...To... =	
ReadAhead =	
Print Statistics =	
Links =	
DBPlus =	
DMSII program titles	
Mirrored Audit	

Installing FileXtract

Complete the following steps to install FileXtract.



Caution: If the DATABridge host database replication software is installed on the mainframe where you want to install FileXtract, see “[Before You Install](#)” on page 11 before completing this step.

Step 1: Determine the Usercode

To determine the usercode, decide where you will be replicating flat files, as follows.

Important: DBEngine is shipped as a privileged program so that it has access to all usercodes.

If you plan to replicate flat files	Then install FileXtract
Under the same usercode where you run FileXtract	Under the same usercode from which you plan to run it.
Under two or more usercodes	With no usercode (same as nonusercoded). If you don't install FileXtract as nonusercoded, establish the DBEngine as a system library (SL). See “ Step 5: Establish DBEngine as a System Library (SL) ” on page 17 for instructions. If you don't establish DBEngine as an SL, you must put a copy of DBEngine under every usercode where you will run FileXtract, or you must file equate each time you run FileXtract.

Step 2: Determine the Pack

After deciding which usercode will contain the FileXtract software, choose either the primary or secondary pack from that usercode's FAMILY substitution statement.

For example, if the usercode has the following substitution

```
FAMILY DISK = PRODPK OTHERWISE DISK
```

then store the FileXtract software on either PRODPK or DISK.

Step 3: Copy the FileXtract Installation WFL

Complete the following steps to copy the installation WFL from the release medium to your host:

- 1 Sign on to the usercode for installing the FileXtract software.

If you are installing FileXtract to a mainframe that has the DATABridge host database replication software, this usercode must be the same as where the DATABridge software currently resides. If you are installing for the first time, this usercode is the one where you intend for the FileXtract files to reside.

- 2 Copy the FileXtract installation WFL from the release medium to your host. Use one of the following methods:

- If you are installing from DVD, use the following command:

```
WFL UNWRAP *WFL/DATABRIDGE/INSTALL AS
WFL/DATABRIDGE/INSTALL OTOF *INSTALL FROM DBXX
(DVD) TO DISK (RESTRICTED = FALSE)
```

where XX is the version of the software with no decimals. For example, DB61.

It is recommended that you install to a privileged usercode. If you install from DVD to a nonprivileged usercode, all object files will be marked as restricted, which causes the installation WFL to pause (in “Step 4: Start the FileXtract Installation WFL.”).

- If you are installing from uploaded files, upload DISKINSTALL and IMAGE files using a file transfer tool capable of binary/image transfers to the usercode where DATABridge is to be installed. Sign on to the usercode and then use the following command:

```
WFL UNWRAP *WFL/DATABRIDGE/INSTALL AS WFL/DATABRIDGE/
INSTALL OTOF DISKINSTALL
```

Step 4: Start the FileXtract Installation WFL

Complete the following steps to start the FileXtract installation WFL:

- 1 From CANDE or another editor, start the installation WFL as follows:

```
START WFL/DATABRIDGE/INSTALL ("FILEXTRACT"
[, "familyname"])
```

where *familyname* is the name of the pack you determined in “Step 2: Determine the Pack” on page 14. *familyname* is optional and is only required if you did not install the WFL to the primary family. If you don’t include the *familyname* parameter, the installation defaults to DISK.

- 2** When prompted, enter the access code (same as license key) located on your DATABridge host DVD package. FileExtract uses the same key as the DATABridge host software. If you want to install an evaluator copy enter the evaluation code located on your DATABridge host DVD package.

If the installation WFL pauses, continue with step **a**.

Otherwise, proceed to “Step 5: Establish DBEngine as a System Library (SL).”

- a** If you installed from DVD to a nonprivileged usercode and the installation WFL is paused, you must unrestrict OBJECT/DATABRIDGE/KEYENTRY, as in this example:

```
RESTRICT -FILE (ENGR) OBJECT/DATABRIDGE/KEYENTRY ON  
PRODUCTION
```

- b** After you have unrestricted OBJECT/DATABRIDGE/KEYENTRY, transmit *mixnumber*OK to the install WFL job.

OBJECT/DATABRIDGE/KEYENTRY now runs.

- c** When prompted, enter the access code (same as license key) located on your DATABridge host DVD package.
- d** After the installation WFL completes, mark *all* of the DATABridge object files as unrestricted by entering the following from either the ODT or from a privileged usercode in MARC:

```
RESTRICT -FILE objectfilename
```

Step 5: Establish DBEngine as a System Library (SL)

This step is optional, but recommended. Establishing DBEngine as an SL ensures that all DATABridge Accessories (as well as other programs) can access the DBEngine without specifying a location for DBEngine. It is recommended that you establish DBEngine as an SL.

Note: If you established DBEngine as an SL in a previous release of DATABridge, you *must* establish DBEngine as an SL for this release of FileXtract. Otherwise, the new Accessories will try to use the old DBEngine.

Additionally, if you want to access DBEngine from several usercodes, but you don't want to install it as a nonusercoded file, you must establish it as an SL.

To establish DBEngine as an SL, enter the following command:

```
SL DBENGINE = (usercode)OBJECT/DATABRIDGE/ENGINE ON  
familyname
```

Installation Results

The following files are copied to the usercode you designated when you installed FileXtract:

File Name	Description
INSTALLS/DBENTERPRISE/ "SETUP.EXE"	Executable file for installing DATABridge Enterprise on a ClearPath PC or Windows server
WFL/DATABRIDGE/ BACKUPTAILORED	Workflow for backing up tailored DMSII software, DESCRIPTION and DMSUPPORT, with the update level as the last node of the file titles.
WFL/DATABRIDGE/COMP	Work flow for compiling a DATABridge sample program or DBSupport
WFL/DATABRIDGE/DBINFO	Work flow for executing the DBInfo Accessory
WFL/DATABRIDGE/INCLUDE/ DBTITLE	Work flow that parses the DMSII database title for all other WFLs
WFL/DATABRIDGE/INSTALL	Work flow for installing DATABridge and FileXtract files
WFL/DATABRIDGE/LISTER	Work flow for executing the DBLister Accessory
WFL/DATABRIDGE/SERVER	Work flow for executing the DBServer Accessory
WFL/DATABRIDGE/SNAPSHOT	Work flow for executing the DBSnapshot Accessory
WFL/DATABRIDGE/SPAN	Work flow for executing the DBSpan Accessory
WFL/FILEBRIDGE/COMP	Work flow for compiling a FileXtract program
WFL/FILEBRIDGE/COBOLTODASDL	Work flow for executing the COBOL- to-DASDL utility
DATA/LOAD/SAMPLE/DATABASE	Data for the sample database that you can use with DATABridge
DATA/ENGINE/SAMPLE/CONTROL	Sample DBEngine parameter file for specifying database-specific DBEngine parameters (SEQDATA file type)

File Name	Description
DATA/ENGINE/CONTROL	CANDE text file that contains the key information for running this release of DATABridge and FileXtract, as well as other parameters for DBEngine operation
DATA/LISTER/SAMPLE/CONTROL	Sample parameter file for the DBLister Accessory (SEQDATA file type)
DATA/SERVER/SAMPLE/CONTROL	Sample parameter file for the DBServer Accessory (SEQDATA file type)
DATA/SERVER/SAMPLE/ FILEBRIDGE/CONTROL	Sample parameter file for the DBServer Accessory (SEQDATA file type)
DATA/GENFORMAT/SAMPLE/ BANKFILESDB/CONTROL	Sample DBGenFormat parameter file for the BANKFILES database
DATA/GENFORMAT/SAMPLE/ LINCLOGDB/CONTROL	Used for linc log tailored support library.
DATA/GENFORMAT/SAMPLE/ NAPFILESDB/CONTROL	Sample DBGenFormat parameter file for the NAPFILES database
DATA/GENFORMAT/SAMPLE/ SYFILESDB/CONTROL	Sample DBGenFormat parameter file for the SYFILES database
DATA/GENFORMAT/SAMPLE/ USERDATADB/CONTROL	Sample DBGenFormat parameter file for the USERDATA database
DATA/FILEBRIDGE/README DATA/API/README	CANDE text file that lists important notices and last minute information about FileXtract
DATA/COBOLTODASDL/SAMPLE/ BANKFILE/FD	FD file for the sample BANKFILE database
DATA/COBOLTODASDL/SAMPLE/ BANKFILE/CONTROL	Sample COBOL-to-DASDL parameter file for the BANKFILE database
DATA/COBOLTODASDL/SAMPLE/ LINCLOGDB/FD	FD file for the sample LINCLog database
DATA/COBOLTODASDL/SAMPLE/ LINCLOGDB/CONTROL	Sample COBOL-to-DASDL parameter file for the LINCLog database
PATCH/DATABRIDGE/PARSER/ DEFINES	Definitions for the parameter file parser
PATCH/DATABRIDGE/SAMPLE/ SUPPORT/STARTUP	Sample startup code that is executed before the DBSupport library freezes

File Name	Description
PATCH/DATABRIDGE/SAMPLE/SUPPORT/VIRTUAL	Sample virtual transform routine
PATCH/DATABRIDGE/SAMPLE/SUPPORT/REFORMAT	Sample INTERNAL REFORMAT routine
PATCH/DATABRIDGE/SAMPLE/SUPPORT/SHUTDOWN	Sample shutdown code that is executed after the DBSupport library thaws
PATCH/DATABRIDGE/SAMPLE/SUPPORT/NAPREFORMAT	Sample reformatting routine for the NAPFilesDB database
PATCH/DATABRIDGE/SAMPLE/SUPPORT/ERRORHANDLER	Sample error manager routine Refer to the <i>DATABridge Programmer's Reference</i> for information about creating a custom error manager routine.
PATCH/DATABRIDGE/SAMPLE/SUPPORT/FORMATADDRESS	Sample transform routine for VIRTUAL data sets
PATCH/DATABRIDGE/SAMPLE/SUPPORT/SYSREFORMAT	Sample patch for reformatting sysfilesdb items
PATCH/DATABRIDGE/COBOLFD/PARSER	Code that parses the COBOL FD file
PATCH/DATABRIDGE/COBOLTODASDL/PARSER	Code that parses the COBOLTODASDL parameter file
OBJECT/DATABRIDGE/SPAN	Object code for DBSpan Accessory
OBJECT/DATABRIDGE/DBINFO	Object code for DBInfo Accessory
OBJECT/DATABRIDGE/ENGINE	Object code for DBEngine library
OBJECT/DATABRIDGE/LISTER	Object code for DBLister Accessory
OBJECT/DATABRIDGE/SERVER	Object code for the DBServer Accessory
OBJECT/DATABRIDGE/SUPPORT	Object code for DBSupport library
OBJECT/DATABRIDGE/SNAPSHOT	Object code for DBSnapshot Accessory
OBJECT/DATABRIDGE/GENFORMAT	Object code for DBGenFormat utility
OBJECT/DATABRIDGE/COBOLSUPPORT	Object code for DBCobolSupport Reader library

File Name	Description
OBJECT/FILEBRIDGE/READER/ BANKFILE	Object code for the sample Reader library for the sample BANKFILE database
OBJECT/FILEBRIDGE/READER/ BICSS	Object code for the sample BICSS log files Reader library
OBJECT/FILEBRIDGE/READER/ COMS	Used for linc log tailored support library
OBJECT/FILEBRIDGE/READER/ DISKFILE	Object code for the sample disk files Reader library
OBJECT/FILEBRIDGE/READER/ LINCLOG	Object code for the sample LINCLOG files Reader library
OBJECT/FILEBRIDGE/READER/ PRINTFILE	Object code for the sample printer backup files Reader library
OBJECT/FILEBRIDGE/READER/ SUMLOG	Object code for the sample system summary log Reader library
OBJECT/FILEBRIDGE/READER/ TTRAIL	Object code for the sample COMS TTrail Reader library
OBJECT/FILEBRIDGE/INITIALIZE	Object code that prepares the layout database control file for FileXtract
OBJECT/FILEBRIDGE/ PATCHDASDL	Object code that patches the layout database DASDL for use by FileXtract
OBJECT/FILEBRIDGE/ COBOLTODASDL	Object code for the COBOL-to-DASDL utility
OBJECT/FILEBRIDGE/ USERDATATODASDL	Reads the USERDATA file layout to generate the DASDL that FileXtract will use.
SYMBOL/DATABRIDGE/SAMPLE/ REFORMAT	Sample source code for writing data item conversion routines in a REFORMAT library (ALGOL file type)
SYMBOL/DATABRIDGE/SAMPLE/ COBOLFORMAT	Sample source code for writing a formatting routine in COBOL (COBOL74 file type)
SYMBOL/DATABRIDGE/SAMPLE/ COBOLTRANSFORM	Sample DATABridge TRANSFORM in COBOL.
SYMBOL/DATABRIDGE/SAMPLE/ EXTRACTADDRESS	Sample source code for a library that will extract addresses and build VIRTUAL records (COBOL file type)

File Name	Description
SYMBOL/DATABRIDGE/ SUPPORT	Source code for DBSupport library (ALGOL file type)
SYMBOL/DATABRIDGE/ INTERFACE	Source code INCLUDE file for the API (ALGOL file type)
SYMBOL/FILEBRIDGE/SAMPLE/ READER/TEXT	Source code for the disk file Reader library
SYMBOL/FILEBRIDGE/SAMPLE/ READER/BICSS	Source code for the BICSS Reader library
SYMBOL/FILEBRIDGE/SAMPLE/ READER/SUMLOG	Source code for the SUMLOG Reader library
SYMBOL/FILEBRIDGE/SAMPLE/ READER/TTRAIL	Source code for the TTRAIL Reader library
SYMBOL/FILEBRIDGE/SAMPLE/ READER/BANKFILE	Source code for the BANKFILE Reader library
SYMBOL/FILEBRIDGE/SAMPLE/ READER/COMS	Source code for the COMS configuration file reader.
SYMBOL/FILEBRIDGE/SAMPLE/ READER/DISKFILE	Source code for the DISKFILE Reader library
SYMBOL/FILEBRIDGE/SAMPLE/ READER/PRINTFILE	Source code for the PRINTFILE Reader library
SYMBOL/FILEBRIDGE/SAMPLE/ READER/LINCLOG	Source code for the LINCLOG Reader library
SYMBOL/FILEBRIDGE/SAMPLE/ READER/USERDATAFILE	The reader for USERDATA which contains user codes and their attributes for the host.
SYMBOL/FILEBRIDGE/SAMPLE/ NAPFILESDB	Source code for the sample flat file NAPFILESDB database
SYMBOL/FILEBRIDGE/SAMPLE/ SYSFILESDB	Source code for the sample flat file SYSFILESDB database
SYMBOL/FILEBRIDGE/SAMPLE/ BANKFILESDB	Source code for the sample flat file BANKDB database
SYMBOL/FILEBRIDGE/SAMPLE/ PRINTFILESDB	Source code for the sample flat file PRINTFILESDB database
SYMBOL/FILEBRIDGE/INITIALIZE	Source code that prepares the layout database control file for FileXtract
SYMBOL/FILEBRIDGE/ PATCHDASDL	Source code for the Patch DASDL utility

File Name	Description
SYMBOL/FILEBRIDGE/ COBOLTODASDL	Source code for the COBOL-to- DASDL utility
SYMBOL/FILEBRIDGE/ USERDATATODASDL	Source code for OBJECT/ FILEBRIDGE/USERDATATODASDL.

Using the Sample Reader Libraries

3

In This Chapter

This chapter explains how to use the sample Reader libraries that come with FileXtract. The following sections are included:

Understanding the Sample Reader Libraries	26
Using the Sample Reader Libraries	34

Understanding the Sample Reader Libraries

The following sample Reader libraries are provided with FileXtract:

- SUMLOG
- TTRAIL
- PRINTFILE
- BICSS
- LINCLOG
- DISKFILE
- BANKFILE
- USERDATA
- TEXT

These Reader libraries are configured to replicate flat files as standard data sets for the given file type. In most cases, you can use these Reader libraries without having to modify them or with very little modification. You will need to modify a Reader library if your flat files contain data in a proprietary format, if new fields have been added to a standard layout, or if you want to replicate additional types of records that the Reader library does not support (for example, a new SUMLOG record type).

The Reader libraries work in conjunction with a physical database that DBEngine uses to determine the layout of the flat files. DBEngine gets all of the information it needs from the DESCRIPTION file for the layout database. The DMSII CONTROL file and the DMSUPPORT library will exist, but the data sets and audit files will not.

A DASDL file is provided for each sample Reader library so that you can create the corresponding layout database. For example, if you want to replicate SUMLOGs, you would use the SYSFILESDB DASDL to create a physical database called SYSFILESDB. Within this database is a logical database called SUMLOG that contains typical layout information for SUMLOG files. If needed, you can modify this information. If you customize the layout information in the logical database, you must make the corresponding changes in the Reader library.

SUMLOG Reader Library

This Reader library reads system summary log files and returns the data records to DBEngine. This Reader library uses the SUMLOG logical database of the SYSFILESDB layout database.

Closed SUMLOG files are typically titled:

```
*SUMLOG/ssss/mmdyy/nnnnn ON familyname
```

where *ssss* is the system serial number, *mmdyy* is the date in month-day-year format, and *nnnnn* is a sequential number.

The Reader option in the Accessory's parameter file should specify the directory and family name where the closed log files are located. The following is an example of the DBServer SOURCE declaration for this Reader library:

```
Source SUMLOG:
%      -----
      Database = SUMLOG of DESCRIPTION/SYSFILESDB ON
          DISK,
      Support  = OBJECT/DATABRIDGE/SUPPORT/SYSFILESDB
          ON DISK,
      Filter   = SUMLOGFILTER,
      Reader using "SUMLOG/= ON DISK"
```

If the Reader option is missing or empty, the SUMLOG Reader library will use the following:

```
"*SUMLOG/ssss/= ON sumlogpack"
```

where *ssss* is the serial number of the system it is running on and *sumlogpack* is the pack containing *SYSTEM/SUMLOG. When all of the closed SUMLOGs have been processed, the Reader library will automatically switch to reading the active SUMLOG.

TTRAIL Reader Library

This Reader library reads COMS Transaction Trail files and returns the data records to DBEngine. This Reader library uses the TTRAIL logical database of the SYSFILESDB layout database.

TTRAIL files are typically titled:

```
*COMS/TTRAIL/dbname/nnnn ON familyname
```

where *dbname* is the database that participates in synchronized recovery and *nnnn* is a sequential number. The TTRAIL for all non-synchronized recovery programs has a database name of TPLIBRARY.

The Reader option in the Accessory's parameter file must specify the directory and family name where the TTRAIL files for a particular database are located. The following is an example of the DBServer SOURCE declaration for this Reader library:

```
Source TTRAIL:
%      -----
      Database = TTRAIL of DESCRIPTION/SYSFILESDB ON
          DISK,
      Support  = OBJECT/DATABRIDGE/SUPPORT/SYSFILESDB
          ON DISK,
      Filter   = TTRAILFILTER,
      Reader using "COMS/TTRAIL/= ON DISK";
```

PRINTFILE Reader Library

This Reader library reads printer backup files and returns the printer file attributes and print lines to DBEngine. This Reader library uses the PRINTFILE logical database of the SYSFILESDB layout database.

The Reader option in the Accessory's parameter file should specify the directory and family name where the print files are located. The following is an example of the DBServer SOURCE declaration for this Reader library:

```
Source LISTER:
%      -----
      Database = REPORTS of DESCRIPTION/PRINTFILESDB ON
          DISK,
      Reader using "DBBD/RUN/LISTER/= ON DISK";
```

If the Reader option is missing or empty, the PRINTFILE Reader library will use:

```
"*BD ON printerbackupack"
```

where *printerbackupack* is the value identified by issuing a DL BACKUP command.

BICSS Reader Library

This Reader library reads BICSS log files and returns the data records to DBEngine. This Reader library uses the UVMSBICSS or IVRBICSS logical database of the NAPFILESDB layout database.

BICSS log files are typically titled:

```
(usercode)BICSSLOG/ACTIVITY/yymmdd/hhmnss ON familyname
```

where *yymmdd* is the date in year-month-day format and *hhmnss* is the time in hours-minutes-seconds format.

The Reader option in the Accessory's parameter file must specify the directory and family name where the log files are located. Optionally, it can also specify the particular application system name that must appear in all records by appending a colon and the LINC application name, such as IVRSYS (see the bolded example below) or UVMS. If you want records from only one application and the log contains records from more than one application, specifying the application name after the colon speeds up the processing.

The following is an example of the DBServer SOURCE declaration for this Reader library:

```
Source IVRBICSS:
%      -----
      Database = IVRBICSS of DESCRIPTION/NAPFILESDB ON
      DISK,
      Support  = OBJECT/DATABRIDGE/SUPPORT/NAPFILESDB ON
      DISK,
      Filter   = BICSSFILTER,
      Reader using "(BICSS)BICSSLOG/ACTIVITY/= ON DISK:
      IVRSYS";
```

DISKFILE Reader Library

This Reader library reads flat files and returns the data records to DBEngine.

When you use the COBOL-to-DASDL utility to create the various structures needed to replicate flat files that use COBOL 01-level file record descriptions (as explained in “[Generating Database Layout from COBOL 01-Level File Record Descriptions](#)” on page 44), the DISKFILE Reader library is the default Reader library.

The DISKFILE Reader library processes disk files in creation timestamp order (see “[Out-of-Order Processing](#)” on page 60 if you need to process a flat file having a creation timestamp prior to what FileXtract has already processed). It assumes that the file it is reading can be extended as long as there is no file with a later creation timestamp. During processing, the creation timestamp of the disk file is used as the timestamp associated with each record.

The Reader option in the Accessory’s parameter file must specify the directory and family name where the disk files are located. The following is an example of the DBServer SOURCE declaration for this Reader library:

```
Reader using "(PROD)DATA/TRANLOG/= ON USERPACK"
```

If you want to change this behavior, you may modify the DISKFILE Reader library or substitute your own Reader library. If you use a name other than OBJECT/FILEBRIDGE/READER/DISKFILE, make sure to specify the title of the Reader library code file in the READER "*codefiletitle*" USING "... " option of the Accessory’s parameter file.

Some programs, such as CANDE and FileXpress®-XST, rewrite the entire file when records are added, deleted, or modified. The (modified) file will have a new creation timestamp. When the DISKFILE Reader searches the disk directory for files to process, it will find the file with a new creation timestamp and will attempt to process it from the beginning. This is likely to cause many “duplicates” errors because it will send each record as an “add” and all of the records from the old file are already present in the client database. To avoid these errors, you must configure the client to reclone every time it is run.

Similarly, if the mainframe application updates records in place, the DISKFILE Reader will not detect the update if it has already processed that file. In this situation, you must also configure the client to reclone every time it is run.

LINCLOG Reader Library

This Reader library reads LINC Activity logs and returns the data records to DBEngine.

LINC log files are typically titled:

```
(usercode) lincsystem/LINCLOG/nnn ON familyname
```

where *nnn* is the Activity log number.

The Reader option in the Accessory's parameter file should specify the directory and family name where the closed log files are located.

In order to generate the LINCLog database so that DATABridge can replicate the log records to the client database, perform the following steps.

- 1 Edit the DATA/COBOLTODASDL/SAMPLE/LINCLOGDB/FD file, which describes the layout of the LINC Activity Log records, to reflect your particular LINC system. (Different LINC implementations sometimes have new fields added to the log record layout.)
- 2 Save the file as DATA/COBOLTODASDL/LINCLOGDB/FD, i.e., without the SAMPLE node.
- 3 Edit the DATA/COBOLTODASDL/SAMPLE/LINCLOGDB/CONTROL file with the proper location for the LINC log files.
- 4 Save the file as DATA/COBOLTODASDL/LINCLOGDB/CONTROL, again without the SAMPLE node.
- 5 Use the COBOL-to-DASDL utility to generate the database:

```
START WFL/FILEBRIDGE/COBOLTODASDL ("LINCLOG")
```

This also generates a file called DATA/SERVER/LINCLOG/CONTROL that you can insert into DATA/SERVER/CONTROL to define the LINCLog SOURCE.

- 6 If you want to filter the log records, declare a filter and generate a tailored DBSupport library using DBGenFormat.
- 7 Use the DATABridge Client or an Accessory such as DBSpan to replicate the LINC Log data just like any other database.

The following is an example of the DBServer SOURCE declaration for this Reader library:

```
Source "LINCLOGFILES":
%
-----
Database = "LINCLOGFILES" OF
          "DESCRIPTION/LINCLOG",
Reader using "(PROD)G2/LINCLOG/= ON AUDIT";
```

Reader processes the disk files in the LINC log file number order (with wraparound from 999 to 1).

BANKFILE Reader Library

This Reader library is a sample flat file Reader library that demonstrates how to write a FileXtract Reader in COBOL. See “[BANKFILE COBOL-to-DASDL Example](#)” beginning on page 41 for an example of how to use this Reader library to replicate a sample flat file called DATA/LOAD/SAMPLE/DATABASE.

USERDATA Reader Library

This Reader library reads the system USERDATA file layout and returns the usercode attributes to DBEngine. It interprets the structure of the system USERDATA file, which contains attributes associated with usercodes. Because the layout of the USERDATA file can vary from site to site, this Reader library generates a DMSII database layout corresponding to your site-specific USERDATA file.

The USERDATA Reader library generates the following:

- The corresponding DASDL layout for a DMSII database for FileXtract
- The corresponding source code for the USERDATA Reader library

To generate and compile the DASDL layout and the USERDATA Reader library, do the following:

```
START WFL/FILEBRIDGE/COMP ("USERDATA")
```

The Reader option in the Accessory's parameter file should specify the directory and family name where the USERDATA file is located. The following is an example of the DBServer SOURCE declaration for this Reader library:

```
Source USERDATA:
%
-----
Database = USERFILE of DESCRIPTION/USERDATADB
Support = OBJECT/DATABRIDGE/SUPPORT/USERDATADB
Filter = USERDATAFILTER
%- Reader using "SYSTEM/USERDATAFILE/othersystem ON
DISK";
```

If the Reader option is missing or empty, the USERDATA Reader library will use:

```
*SYSTEM/USERDATAFILE ON userdatapack
```

where *userdatapack* is disk location specified by the DL USERDATA command.

TEXT Reader Library

This is a skeleton of a Reader library written in ALGOL. It is intended to be used when writing a custom ALGOL Reader library. See [Chapter 5, “Creating a Custom Reader Library,”](#) beginning on page 51 for information about writing a custom Reader library.

Using the Sample Reader Libraries

Complete the following steps to use a sample Reader library to replicate SUMLOGs, COMS TTrail files, printer backup files, BICSS log files, or BankDB sample data files.

For flat files using a COBOL FD, including LINC Activity logs created by LINC systems, see [“Understanding the COBOL-to-DASDL Utility”](#) beginning on page 40.

- 1 Review the sample DASDL for the layout database associated with the appropriate Reader library to make sure it contains the layouts you desire.

For this Reader library	Review this sample DASDL
SUMLOG or TTRAIL	SYMBOL/FILEBRIDGE/SAMPLE/SYSFILESDB
PRINTFILE	SYMBOL/FILEBRIDGE/SAMPLE/PRINTFILESDB
BICSS	SYMBOL/FILEBRIDGE/SAMPLE/NAPFILESDB
BANKFILE	SYMBOL/FILEBRIDGE/SAMPLE/BANKFILESDB

If you want to make changes in the layout database, copy the sample DASDL using the following command, then edit and save the new DASDL:

```
COPY SYMBOL/FILEBRIDGE/SAMPLE/dbname AS SYMBOL/  
FILEBRIDGE/dbname
```

- 2 Use the following command to create the layout database that DBEngine will use to determine the layout of the flat files.

Note: This WFL compiles layout tables for each flat file in the following layout databases. No updates are actually done to this layout database. DBEngine gets all of the information it needs from the DESCRIPTION file. The DMSII CONTROL file and the DMSUPPORT library will exist, but the data sets and audit files will not.

```
START WFL/FILEBRIDGE/COMP ("dbname")
```

Where <i>dbname</i> is	For
SYSFILESDB	SUMLOG or TTRAILReader libraries The SYSFILESDB physical database contains the SUMLOG and TTRAIL logical databases.
PRINTFILESDB	PRINTFILE Reader library The PRINTFILESDB physical database contains the PRINTFILE logical database.
NAPFILESDB	BICSS Reader library The NAPFILESDB physical database contains the UVMSBICSS and IVRBICSS logical databases.
BANKFILESDB	BANKFILE Reader library The BANKFILESDB physical database contains the BANKDB logical database.

- 3 View the Reader library source and verify that it handles all of the record types you will be replicating.

If it does not, modify the Reader library, save the source without the SAMPLE node (for example, save SYMBOL/FILEBRIDGE/SAMPLE/READER/SUMLOG as SYMBOL/FILEBRIDGE/READER/SUMLOG) and then use the following command to recompile the Reader library:

```
START WFL/FILEBRIDGE/COMP ("reader")
```

where *reader* is either SUMLOG, TTRAIL, PRINTFILE, BICSS, or BANKFILE.

Note that you can compile the database and associated Reader library in a single job by starting WFL/FILEBRIDGE/COMP with one of the following parameters:

"NAPFILES"
"BANKFILES"
"PRINTFILES"
"SYSFILES"
"USERDATA"
"LINCLOGS"

(Before starting WFL/FILEBRIDGE/COMP for the LINCLog database and Reader, you must perform steps 1 - 4 in [“LINCLOG Reader Library”](#) on page 31.)

Note: If you are using a layout database that does not require a tailored DBSupport library, such as PRINTFILESDB, you are ready to configure the READER declaration for the Accessory you will use to replicate the flat file. See the next section, [“What to Do Next”](#) for instructions.

- 4 Copy the DBGenFormat parameter file as follows:

```
COPY DATA/GENFORMAT/SAMPLE/dbname/CONTROL AS DATA/  
GENFORMAT/dbname/CONTROL
```

where *dbname* is SYSFILESDB, USERDATADB, NAPFILESDB, or BANKFILESDB.

- 5 Compile the DBSupport library for the database as follows:

```
START WFL/DATABRIDGE/COMP ("SUPPORT", "dbname")
```

where *dbname* is SYSFILESDB, USERDATADB, NAPFILESDB, or BANKFILESDB.

This command results in the following tailored support library:

```
OBJECT/DATABRIDGE/SUPPORT/dbname
```

where *dbname* is SYSFILESDB, USERDATADB, NAPFILESDB, or BANKFILESDB.

What to Do Next

The Reader library is ready to clone the flat files. Do one of the following based on which DATABridge Accessory you are going to use:

For	Do this
DBServer	<ol style="list-style-type: none">1 If you already have a DBServer parameter file, insert the appropriate FileXtract SOURCE declarations from DATA/SERVER/SAMPLE/FILEBRIDGE/CONTROL at the end of DATA/SERVER/CONTROL. Otherwise, copy the sample FileXtract parameter file DATA/SERVER/SAMPLE/FILEBRIDGE/CONTROL as DATA/SERVER/CONTROL.2 Run DBServer. Refer to the <i>DATABridge Host Administrator's Guide</i> for instructions.3 Run the Client with the DEFINE command. Refer to the <i>DATABridge Clients Administrator's Guide</i> for instructions.
Other Accessories	Define the READER option in the Accessory's parameter file and then run the Accessory with the name of the logical database as the second parameter. Refer to the <i>DATABridge Host Administrator's Guide</i> for instructions.

Using the COBOL-to-DASDL Utility

4

In This Chapter

This chapter explains how to use the COBOL-to-DASDL utility to replicate flat files that use COBOL 01-level file record descriptions. The following sections are included in this chapter:

Understanding the COBOL-to-DASDL Utility	40
BANKFILE COBOL-to-DASDL Example	41
Generating Database Layout from COBOL 01-Level File Record Descriptions	44
Generating Database Layout from LINC Activity Log Files	48

Understanding the COBOL-to-DASDL Utility

The COBOL-to-DASDL utility allows you to generate a FileXtract layout database from COBOL 01-level file record descriptions. In addition to the DASDL for the new database, the COBOL-to-DASDL utility also generates the appropriate DBServer SOURCE declaration that you insert into the DBServer parameter file so that DATABridge Clients can access the replicated flat files.

- For normal disk files, the DISKFILE Reader library is used to read the flat files that are being replicated.

For a description of the DISKFILE Reader library, see “[DISKFILE Reader Library](#)” on page 30.

Otherwise, see “[Generating Database Layout from COBOL 01-Level File Record Descriptions](#)” on page 44.

Note: You must have or create a COBOL file containing the 01-level file record descriptions for the flat files you want to replicate. A sample COBOL file (DATA/COBOLTODASDL/SAMPLE/LINCLOG/FD) containing a layout of the LINC log is installed with the FileXtract software.

- For LINC log activity files, the LINCLOG Reader library is used to read the flat files that are being replicated.

For a description of the LINCLOG Reader library, see “[LINCLOG Reader Library](#)” on page 31.

Otherwise, see “[Generating Database Layout from LINC Activity Log Files](#)” on page 48.

Example

See “[BANKFILE COBOL-to-DASDL Example](#)” beginning on page 41 for an example of how to use the COBOL-to-DASDL utility to replicate a sample flat file called DATA/LOAD/SAMPLE/DATABASE. It is recommended that you complete this example before using the COBOL-to-DASDL utility for your own disk files.

BANKFILE COBOL-to-DASDL Example

This example illustrates how to use the COBOL-to-DASDL utility to replicate a sample flat file called DATA/LOAD/SAMPLE/DATABASE. Everything you need to run this example is installed with FileXtract.

- 1 View DATA/LOAD/SAMPLE/DATABASE and familiarize yourself with the records in this flat file.
- 2 View and familiarize yourself with DATA/COBOLTODASDL/SAMPLE/BANKFILE/FD, which is the COBOL FD for DATA/LOAD/SAMPLE/DATABASE.
- 3 Copy the BANKFILE parameter file as follows:

```
GET DATA/COBOLTODASDL/SAMPLE/BANKFILE/CONTROL AS DATA/  
COBOLTODASDL/BANKFILE/CONTROL
```

- 4 Define the SOURCE.

The SOURCE name will be used as the DBServer SOURCE name as well as the name of logical database that the COBOL-to-DASDL utility creates.

Note that this step has already been completed for you in this sample control file, as follows:

```
DEFINE Source COBOLBankFile USING "DATA/LOAD/SAMPLE/  
DATABASE" FROM FD IN "DATA/COBOLTODASDL/SAMPLE/  
BANKFILE/FD";
```

- 5 Declare the data sets to be generated from the various flat files.

Note that this step has already been completed for you in this sample control file. The following are the data sets that have been defined:

```
Dataset Comments  
uses INPUT-DATA-REC  
where COMMENT-REC; % 88-level item
```

```
Dataset BANK  
uses INPUT-BANK % => use the INPUT-BANK REDEFINE  
where BANK-TYPE and (BANK-ID not greater than 5000  
or BANK-NAME ^= SPACES); % a more complex expression
```

```
Dataset BRANCH  
uses INPUT-BRANCH, DATA-TYPE  
where BRANCH-TYPE;
```

```
Dataset CUSTOMER
  uses INPUT-CUSTOMER
  where CUSTOMER-TYPE;
```

```
Dataset TELLER
  uses INPUT-TELLER
  where TELLER-TYPE;
```

```
Dataset ACCOUNT
  uses INPUT-ACCOUNT
  where ACCOUNT-TYPE;
```

```
Dataset HISTORY
  uses INPUT-HISTORY
  where HISTORY-TYPE;
```

- 6** Save the parameter file.
- 7** Run the COBOL-to-DASDL utility using the following command:

```
START WFL/FILEBRIDGE/COBOLTODASDL ("BANKFILE")
```

The COBOL-to-DASDL utility processes the COBOL file description, generates a DASDL called SOURCE/FILEBRIDGE/COBOLTODASDL/BANKFILE, and generates a DBServer parameter file fragment called DATA/SERVER/BANKFILE/CONTROL. Then, the WFL compiles the resulting database and runs the FileXtract Initialize utility to prepare it for DATABridge.

- 8** Update the DBServer parameter file with the generated DBServer parameter file fragment, as follows:

- e** Transmit the following command:

```
GET DATA/SERVER/CONTROL
```

- f** Transmit the following command:

```
INSERT DATA/SERVER/BANKFILE/CONTROL AT END
```

The DBServer parameter file is appended with the BANKFILE DBServer parameter file fragment, which defines the SOURCE, database, and data file title to use to replicate the BANKFILE flat file.

- 9** Run the DATABridge Client with the DEFINE command to define the COBOLBankFile source. Then run it with the GENERATE command and finally with the PROCESS command.

The DATABridge Client contacts DBServer to replicate the BANKFILE flat file. Refer to the *DATABridge Clients Administrator's Guide* for instructions on running the DATABridge Client.

Generating Database Layout from COBOL 01-Level File Record Descriptions

The sample control file for the COBOL-to-DASDL utility (DATA/COBOLTODASDL/SAMPLE/BANKFILE/CONTROL) contains sample entries based on a flat file called DATA/LOAD/SAMPLE/DATABASE. This file illustrates how to declare a DBServer SOURCE and generate the associated layout database from COBOL 01-level file record descriptions. As you complete the following steps, remove, modify, or comment out the sample entries.

Note: The COBOL-to-DASDL utility converts all non-computational numeric fields to type ALPHA.

Complete the following instructions to replicate flat files that use COBOL 01-level file record descriptions.

- 1 Copy the COBOL-to-DASDL parameter file as follows:

```
Get DATA/COBOLTODASDL/SAMPLE/BANKFILE/CONTROL AS DATA/  
COBOLTODASDL/dbname/CONTROL
```

where *dbname* is the name you want to use for the layout database that will be generated.

- 2 Define the DBServer SOURCE declaration as follows:

```
DEFINE SOURCE sourcename USING "filename"  
FROM FD IN "COBOLsourcefile";
```

Notes:

- *sourcename* will be used as the name of the logical database that the COBOL-to-DASDL utility creates. DBSpan and DBSnapshot access flat file data using the logical database.
- *dbname* must be different than *sourcename* and *flatfiledirectory*. If you enter the same name for SOURCE and database, an error message appears.
- *filename* can be a file or a directory. Note that the DISKFILE Reader library treats a file as both a file and a directory. Therefore, that file and all files under that file node will be replicated.
- *COBOLsourcefile* does not have to contain the actual FD, but it must contain the 01-level file record layout(s). However, it cannot contain declarations other than the FD and the 01-level file record layouts. If the FD is in the main COBOL program source file, you must create a new COBOL file containing just the disk file layout and use that file's title for *COBOLsourcefile*.

For example:

```
DEFINE SOURCE COBOLBankFile USING "DATA/LOAD/SAMPLE/  
  DATABASE" FROM FD IN "DATA/COBOLTODASDL/SAMPLE/  
  BANKFILE/FD";
```

In addition to using this information to create the DASDL, the COBOL-to-DASDL utility uses the SOURCE declaration to create a DBServer parameter file fragment (DATA/SERVER/*dbname*/CONTROL) that you insert into the DBServer control file.

- 3 Declare the data sets to be generated from the various flat files using the following syntax.

Note: When you run the COBOL-to-DASDL utility, it automatically encloses the SOURCE and logical database names in “quotation marks” in the generated parameter file fragment for DBServer. This feature allows you to use hyphens (-) in SOURCE names, and the SOURCE names can be the same as a parameter file keyword.

```
DATASET datasetname USES identifier_list [WHERE  
  boolean_expression]
```

For example:

```
DATASET BANK USES INPUT-BANK WHERE BANK-TYPE and (BANK-ID
  not greater than 5000 or BANK-NAME ^= SPACES);
```

The USES clause tells the Reader library what data items to use when there are REDEFINES. Note that successive 01s under an FD are implicit REDEFINES.

The WHERE clause indicates the condition that must be true of all records that belong in that data set. The WHERE clause can be omitted if all records belong in a single data set. The WHERE clause must evaluate to a Boolean expression (true or false).

The Boolean expression can consist of relational expressions, connected by AND or OR, using the normal COBOL relational operators: GREATER THAN, >, GTR, LESS THAN, <, LSS, EQUAL TO, =, EQL, GREATER THAN OR EQUAL TO, >=, GEQ, LESS THAN OR EQUAL TO, <=, LEQ, ^=, !=, <>, NEQ. The left side of a relational expression must be a data item name and the right side must be either an integer, a double-quoted string, a hex string, SPACE, or SPACES. An 88-level condition name can also be used as a relational expression.

- 4 Save the COBOL-to-DASDL parameter file.
- 5 Run the COBOL-to-DASDL utility using the following command:

```
START WFL/FILEBRIDGE/COBOLTODASDL ("dbname")
```

where *dbname* is the name of the layout database you assigned in [step 1](#).

The COBOL-to-DASDL utility processes the COBOL file description, generates the DASDL called SOURCE/FILEBRIDGE/COBOLTODASDL/*dbname*, and generates a DBServer parameter file fragment called DATA/SERVER/*dbname*/CONTROL. Then, the WFL compiles the resulting database and prepares it for use with FileXtract.

What to Do Next

Do one of the following based on which DATABridge Accessory you are going to use:

For	Do this
DBServer	<p>Insert the DBServer parameter file fragment (DATA/SERVER/<i>dbname</i>/CONTROL) into the DATA/SERVER/CONTROL file using the following steps:</p> <ol style="list-style-type: none">1 Get DATA/SERVER/CONTROL.2 Transmit the following command: <pre>INSERT DATA/SERVER/<i>dbname</i>/CONTROL AT END</pre>3 Save the file. <p>The DBServer parameter file is appended with the DBServer parameter file fragment, which defines the SOURCE, database, and Reader library to use to replicate the flat files.</p> <p>After you have updated the DBServer parameter file, run the Client with the DEFINE, GENERATE, and PROCESS commands. Refer to the <i>DATABridge Clients Administrator's Guide</i> for instructions.</p>
Other Accessories	<p>The flat file is ready to be cloned. For DBSpan and DBSnapshot, start the associated WFL with the layout database and logical database (SOURCE) names to create the Accessory's parameter file. For example:</p> <pre>START WFL/DATABRIDGE/SPAN ("BANKFILE", "COBOLBANKFILE")</pre> <p>Define the READER options in the Accessory's parameter file and then run the Accessory. Refer to the <i>DATABridge Host Administrator's Guide</i> for instructions.</p>

Generating Database Layout from LINC Activity Log Files

To generate the LINCLOG database so that DATABridge can replicate the log records to the client database, follow these steps:

- 1 Edit the DATA/COBOLTODASDL/SAMPLE/LINCLOG/FD file, which describes the layout of the LINC Activity Log records, to reflect your particular LINC system. (Different LINC implementations sometimes have new fields added to the log record layout.)
- 2 Save the file as DATA/COBOLTODASDL/LINCLOG/FD, i.e., without the SAMPLE node.

```
SAVE AS DATA/COBOLTODASDL/LINCLOG/FD
```

- 3 Edit the DATA/COBOLTODASDL/SAMPLE/LINCLOG/CONTROL file so that it has the proper location for the LINC log files.

```
DEFINE SOURCE sourcename USING "directoryname"  
FROM FD IN "COBOLsourcefile"
```

Notes:

- *sourcename* will be used as the name of the logical database that the COBOL-to-DASDL utility creates. DBSpan and DBSnapshot access flat file data using the logical database.
- *dbname* must be different than *sourcename* and *directoryname*. If you enter the same name for SOURCE and database, an error message appears.
- *COBOLsourcefile* does not have to contain the actual FD, but it must contain the 01-level file record layout(s).

- 4 Save the file as DATA/COBOLTODASDL/LINCLOG/CONTROL, again without the SAMPLE node.

```
SAVE AS DATA/COBOLTODASDL/LINCLOG/CONTROL
```

- 5** Use the COBOL-to-DASDL utility to generate the database and compile the Reader:

```
START WFL/FILEBRIDGE/COMP ("LINCLOGS")
```

In addition to generating the database (DASDL called SOURCE/FILEBRIDGE/COBOLTODASDL/LINCLOG), the COBOL-to-DASDL utility also generates a file called DATA/SERVER/LINCLOG/CONTROL that you can insert into DATA/SERVER/CONTROL to define the LINCLOG SOURCE.

- 6** If you want to filter the log records, declare a filter and generate a tailored DBSupport library using DBGenFormat. (See the *DATABridge Host Administrator's Guide* for instructions.)

What to Do Next

Do one of the following based on which DATABridge Accessory you are going to use:

For	Do this
DBServer	<p>Insert the DBServer parameter file fragment (DATA/SERVER/<i>dbname</i>/CONTROL) into the DATA/SERVER/CONTROL file using the following steps:</p> <ol style="list-style-type: none"> 1 Get DATA/SERVER/CONTROL. 2 Transmit the following command: <pre>INSERT DATA/SERVER/<i>dbname</i>/CONTROL AT END</pre> 3 Save the file. <p>The DBServer parameter file is appended with the DBServer parameter file fragment, which defines the SOURCE, database, and Reader library to use to replicate the flat files.</p> <p>After you have updated the DBServer parameter file, run the Client with the DEFINE, GENERATE, and PROCESS commands. Refer to the <i>DATABridge Clients Administrator's Guide</i> for instructions.</p>
Other Accessories	<p>The flat file is ready to be cloned. For DBSpan and DBSnapshot, start the associated WFL with the layout database and logical database (SOURCE) names to create the Accessory's parameter file. For example:</p> <pre>START WFL/DATABRIDGE/SPAN ("LINCLOG", "LINCLOGFILES")</pre> <p>Define the READER options in the Accessory's parameter file and then run the Accessory. Refer to the <i>DATABridge Host Administrator's Guide</i> for instructions.</p>

Creating a Custom Reader Library

5

In This Chapter

This chapter explains how to write a custom FileXtract Reader library. The following sections are included in this chapter:

Understanding When to Create a Custom Reader Library	52
Creating a Custom Reader Library	53

Understanding When to Create a Custom Reader Library

You must create a custom Reader library in either of the following situations:

- The flat files that you want to replicate contain information in a “proprietary” format (for example, a flat file that requires reading the *n*th word to find the address of the next record).
- The sample Reader library does not meet your needs. For example, you want to replicate additional types of SUMLOG records.

Creating a Custom Reader Library

The easiest way to create a custom Reader library is simply to customize one of the sample Reader libraries to meet your needs. If this is not possible, complete the following steps to create a custom Reader library.

- 1 Describe the file layouts in either DASDL or COBOL FD format.

Note: If you use COBOL FD format, use the COBOL-to-DASDL utility to generate the DASDL. See for instructions.

Append a VERIFY clause to each data set that indicates when a flat file record is considered part of that data set.

The DASDL must group the data sets into a logical database, as follows:

```
Logical_dbname ["reader [; file_spec]"] DATABASE
  (dataset_list)
```

where *reader* is the title of the Reader library object code and *file_spec* is a file or directory containing the flat files. Note that the semicolon is required before the *file_spec*, if present.

- 2 Write the Reader library source code in ALGOL or COBOL.

Use the TEXT, SUMLOG, or TTRAIL Reader library as an example if you are writing the Reader library in ALGOL. Use the BANKFILE Reader library as an example if you are writing the Reader library in COBOL.

The procedure called in the Reader library must be named FileRead. Use DBFileReaderHead to define the procedure in ALGOL. Use \$FEDLEVEL = 5, \$ SET LIBRARYPROG, and set PROGRAM-ID to FILEREAD for COBOL. FileRead is passed two arrays—the FileInfo structure and FileRecord for the resulting record image.

- 3 Perform required initialization the first time FileRead is called. This might include linking to DBEngine, getting the structure index for the data sets, identifying the location of the flat files, etc.

Note: Linking to DBEngine, whether explicit or implicit, must not occur before the first call of FileRead.

- 4 Configure the Reader library to read the file, filter records, copy the data to the FileRecord array, and set items in FileInfo such as the structure index, file record location, timestamp, etc.

Set the appropriate structure index, as defined in the *DATABridge Programmer's Reference*, in the FileInfo array to determine the data set to which a record belongs. Note that by setting the structure index to 0 in the FileInfo array, you can rely on the DASDL VERIFY clauses to determine the data set to which a record belongs; however, this is not the preferred method because it is slower.

Note: In addition to filtering and formatting done by the Reader library, DBSupport can also provide filtering and formatting. See the *DATABridge Host Administrator's Guide* for more information about DBSupport.

Troubleshooting

A

In This Appendix

This appendix contains troubleshooting information. The following sections are included in this appendix:

General Troubleshooting Procedures	56
Error Messages	58

General Troubleshooting Procedures

If you have problems using FileXtract, complete the following steps:

- 1 Make sure your system meets the requirements necessary to use the product. See “[Required Hardware and Software](#)” on page 10 for this information.
- 2 Make sure the READER option in the parameter file has the correct Reader library and flat file location. If you are using DBServer, verify that the SOURCE declaration is pointing to the correct logical database. If you are using DBSpan or DBSnapshot, make sure you start the WFL with the correct logical database name as the second parameter. For example, START WFL/DATABRIDGE/SPAN ("SYSFILESDB", "SUMLOG"). For information about the Reader libraries, see one of the following:
 - “[Using the Sample Reader Libraries](#)” beginning on page 34
 - “[Generating Database Layout from COBOL 01-Level File Record Descriptions](#)” beginning on page 44
 - “[Generating Database Layout from LINC Activity Log Files](#)” beginning on page 48
 - “[Creating a Custom Reader Library](#)” beginning on page 53
- 3 Check your set up, as follows:
 - Is the DATABridge Accessory running? Check the mix to make sure. Refer to the *DATABridge Host Administrator's Guide* for instructions.
 - Is the READER declaration for the DATABridge Accessory correct? Refer to the *DATABridge Host Administrator's Guide* for instructions.
 - Does the source name for the DEFINE command for the DATABridge Client match the SOURCE name for DBServer?
- 4 Resolve any errors that you receive.

If you are receiving error messages you don't understand, see “[Error Messages](#)” beginning on page 58 for help resolving error these messages.
- 5 If you cannot identify and solve the problem without assistance, contact your product distributor. Call from a location where you have access to the problem mainframe.

- 6** Troubleshoot the problem using information available from Attachmate Technical Support.

<http://www.attachmate.com/en-US/Support/>

This service directly links you to our internal help desk system, 24 hours a day, 7 days a week.

- 7** Contact Attachmate Technical Support:

<http://support.attachmate.com/contact/>

Error Messages

The following are messages that might be generated by the COBOL-to-DASDL utility. You might receive additional messages depending on which DATABridge Accessory you are using. Refer to the *DATABridge Host Administrator's Guide* for information about messages not listed in the following table.

Number	Message	Description
5001	Data item <i>name</i> was not found	The specified name was not found in the COBOL file description.
5002	Data item <i>name</i> is not an 88-level item	The data item was used in the WHERE clause of the DATASET in the parameter file without a relational operator (such as = or >). Only 88-level items can be used as a Boolean value. All others must be part of a relational expression, for example, BANK-ID > 5000.
5003	Unrecognized VALUE type for <i>entry</i>	The right side of a relational expression contains a literal that COBOL-to-DASDL doesn't support. Change the expression to use one of the supported literals listed in Chapter , "Using the COBOL-to-DASDL Utility," beginning on page 39.
5004	Error on <i>linenumber</i> of <i>filename</i>	There is an error in the specified line of the specified file.
5005	Name <i>name</i> is too long (> <i>nn</i>)	The specified name must be <i>nn</i> characters or less.
5006	syntax error message	Either the parameter file or the file description file contains a syntax error as described in the message.
5007	Source name and database name must be different: <i>name</i>	The SOURCE name and the layout database name cannot be the same. Choose a different name for the SOURC <i>name</i> .

Out-of-Order Processing

B

In This Appendix

This appendix explains how to update your relational database created by the DATABridge Client when FileXtract processes files out-of-order. The following section is included in this appendix:

Out-of-Order Processing 60

Out-of-Order Processing

Sometimes, you might want to load a great deal of flat file data that has been archived, and the archived data is not available in the proper sequence. For example, you might have a set of backup tapes with SUMLOGs for several days and one set of tapes in the middle of the sequence is currently “on loan.”

The SUMLOG Reader library (and other Reader libraries) rely on the timestamp from StateInfo to locate the current file and the ABSN concatenated with INX to determine the record number.

In general you want to keep two audit locations—one for the “real” (relatively current) location, and another for the archive-catch-up location.

The following is one approach to handling out-of-order processing. This procedure is for processing archived log files sequentially. Processing archived log files discontinuously requires additional techniques, which are not explained in this guide.

Before Processing

On the host, define a duplicate SOURCE in DBServer (call it SUMLOGARCHIVE) and add a STOP BEFORE *timestamp* condition to it, where the *timestamp* is the timestamp of the oldest record that has already been loaded using the regular SUMLOG source. (This will prevent picking up the same files twice, including the active SUMLOG.) The SUMLOGs for this SOURCE could be in a different directory to avoid confusion.

On the client, you need the two audit locations and a way to switch them back and forth. Create a new table (AudLocSwitch) to hold the variable parts of the audit location. The variable parts are: AFN, ABSN, INX, and timestamp. Also define a text column in AudLocSwitch called “*source_name*.”

Create a row for each data set in AudLocSwitch with all zeros for the audit location and *source_name* = "SUMLOG". Create a similar row for each data set having *source_name* = "SUMLOGARCHIVE."

**Processing
Archived
SUMLOGs**

Complete the following steps to process archived SUMLOG files:

- 1 Copy the audit locations from the data sets table into the AudLocSwitch rows for `source_name = "SUMLOG"`.
- 2 Copy the AudLocSwitch rows for `source_name = "SUMLOGARCHIVE"` into the data sets table.
- 3 Change `data_source` in the `datasources` table and the `datatables` table to `"SUMLOGARCHIVE"`.
- 4 Run the following command:

```
dbutility process SUMLOGARCHIVE
```

**Processing
Current SUMLOGs**

Complete the following steps to switch back to processing the current SUMLOGs:

- 1 Copy the audit locations from the data sets table into the AudLocSwitch rows for `source_name = "SUMLOGARCHIVE"`.
- 2 Copy the AudLocSwitch rows for `source_name = "SUMLOG"` into the data sets table.
- 3 Change `data_source` in the `datasources` table and the `datatables` table to `"SUMLOG"`.
- 4 Run the following command:

```
dbutility process SUMLOG
```

By maintaining separate audit locations and using the STOP BEFORE condition, you can continue to copy in old SUMLOGs and process them and alternately switch back to processing the current SUMLOGs using the other SOURCE.

If the old files are loaded out of order, you must manually remove the processed files and reset the audit location in the AudLocSwitch rows for `"SUMLOGARCHIVE"` to 0 before doing the process command.

Glossary

This glossary includes terms that are unique to FileXtract and DATABridge as well as terms that are standard for DMSII databases. Complete, detailed definitions for ClearPath NX/LX, A Series, and DMSII terms can be found in Unisys documentation. The purpose of this glossary is to explain how these terms relate to FileXtract.

- Accessories** DATABridge Accessories access the services in DBEngine and DBSupport. Some of the Accessories provided with DATABridge are as follows:
- DBServer, which provides communication and DMSII database replication services to DATABridge Clients on remote database systems.
 - DBSpan, which produces a replication of one or more data sets into flat sequential disk files. DBSpan updates the cloned flat files by appending the changes to the end of the flat files (unlike DBSnapshot, which replaces the changed records).
 - DBSnapshot, which produces a one-time replication of one or more data sets into flat sequential disk files or tape. DBSnapshot clones the selected data sets each time you run it.

- DBInfo, which produces a report of your DMSII database timestamps, updates levels, DMSII release levels, etc.
- DBLister, which produces a report of the layout of the structures in your DMSII database, including structure numbers and key sets.

cloning

Cloning is the process of generating a complete snapshot of a data set to another file. Cloning creates a static picture of a dynamic database. DATABridge uses the DMSII data sets and the audit trail to ensure that the cloned data represents a snapshot of the data sets at a quiet point, even though other programs may be updating the database concurrently. DATABridge clones only those data sets you specify.

Cloning is one phase of the database replication process. The other phase is tracking, which is the integration of database changes since the cloning. For more details, see the definition for tracking.

DATABridge Accessories are available for cloning, as follows:

- DATABridge Clients—perform an initial clone of a flat file and then subsequent tracking of the changes made to the flat files. All DATABridge Clients use the DATABridge Server.
- DBSnapshot Accessory—a batch method that provides a one-time snapshot (extraction) only.
- DBSpan Accessory—a dynamic method that provides a one-time extraction and ongoing tracking.

Control file

The DMSII control file is the runtime analog of the DESCRIPTION file. The DESCRIPTION file is updated only when you compile a modified DASDL. The control file controls database interlock. It stores audit control information and verifies that all database data files are compatible by checking the database timestamp, version timestamp, and update level. The control file is updated each time anyone opens the database for updates. The control file contains timestamps for each data set (when the data set was defined, when the data set was updated). It contains parameters such as how much memory the Accessroutines can use and titles of software such as the DMSUPPORT library (DMSUPPORT/*dbname*.)

DATABridge uses the control file for the following information:

- Timestamps
- AFN for the current audit file and ABSN for the current audit block
- Audit file packname
- INDEPENDENTTRANS option
- data set packnames
- Database usercode

DASDL Data and structure definition language—This is the language that defines DMSII databases. The DASDL must be compiled to create a DESCRIPTION file.

data set A file (structure) in DMSII in which records are stored. It is similar to a table in a relational database. You can select the data sets you want to store in your replicated database.

DBEngine DBEngine is a host library program that provides services (entry points) for DATABridge Accessories to retrieve data definition and data records for replication.

DBEngine reads the DESCRIPTION file, the control file, the database data files (via the DMSII Access routines), and initializes DATABridge Plus to read the current audit file.

DBEngine initializes when DATABridge Accessories call it, and one copy is spawned for each Accessory. DBEngine shares the same code stack as other copies of the engine, but it does not share the same data stack.

DBSupport DBSupport provides formatting, translation, and filtering to the DATABridge Accessories. Once an Accessory receives data from DBEngine, the Accessory can call a filtering routine in DBSupport to determine if the data should be replicated, and if so, it passes the data to DBSupport for formatting.

extraction The process of reading through a data set sequentially and writing those records to a file (either a client database or flat file).

replication The process of cloning a flat file and then tracking the changes that occur to that flat file.

Glossary

set An index into a data set. A set has an entry (key + pointer) for every record in the data set.

structure A data set, set, subset, access, or remap. Each structure has a unique number called the structure number.

tracking Retrieving only the changes from the audit file to apply to the replicated database. Tracking is an ongoing process for propagating changes made to records in the DMSII primary database to the replicated database. The DBSpan and DBServer Accessories perform cloning as well as tracking.

Tracking is one phase of the database replication process. The other phase is cloning. For more details, see [“cloning”](#) on page 64.

Index

A

Abbreviations [viii](#)
Access code [16](#)
Accessories, definition [61](#)
AudLocSwitch [58, 59](#)

B

BANKFILE Reader library
 COBOL-to-DASDL example [39–41](#)
 overview [30](#)
 selecting [7](#)
 using [32–35](#)
BICSS Reader library
 overview [26–27](#)
 selecting [6](#)
 using [32–35](#)
Brackets, documentation conventions [vii](#)

C

Caution icon, use of [vii](#)
Cloning, definition [62](#)
COBOL-to-DASDL utility [42–45](#)
 example [39–41](#)
 overview [38](#)
 SOURCE declaration [42](#)
 using [38–47](#)
COMS Transaction Trail Reader library (see
 TTRAIL Reader library)

Control file, definition [62](#)
Conventions [vii, vii, vii, vii, vii](#)
Customer support [55](#)

D

DASDL
 BANKFILESDB [32](#)
 definition [63](#)
 NAPFILESDB [32](#)
 PRINTFILESDB [32](#)
 provided with FileXtract [24](#)
 SYSFILESDB [32](#)
 using COBOL-to-DASDL utility [37](#)
Data and Structure Definition Language (see
 DASDL)
Data items
 case-sensitivity [vii](#)
Data sets
 case-sensitivity [vii](#)
 declaring [43](#)
 definition [63](#)
DATABridge
 accessories [35, 45, 47, 61](#)
 determining the pack for installation [14](#)
 terminology [61–64](#)
DATABridge components
 DBEngine [5](#)
 DBServer [5, 7, 43, 46](#)

Index

- DBSnapshot 5, 7, 43, 46
 - DBSpan 5, 7, 43, 46
 - DBSupport library 5
 - DATABridge Engine (see DBEngine)
 - DATABridge host software, documentation ix
 - DBEngine
 - definition 63
 - description 5
 - registering as system library 14, 17
 - DBServer
 - COBOL-to-DASDL source declaration 43, 46
 - declaring SOURCE and READER options 7
 - description 5
 - parameter file fragment 45, 47
 - DBSnapshot
 - COBOL-to-DASDL utility 43, 46
 - declaring READER option 7
 - description 5
 - DBSpan
 - COBOL-to-DASDL utility 43, 46
 - declaring READER option 7
 - description 5
 - DBSupport library
 - definition 63
 - overview 5
 - DISKFILE Reader library
 - overview 28
 - selecting 6
 - using 42–45
 - DMSUPPORT library, related to control file 62
 - Documentation conventions vii
 - Documentation related to using DATABridge
 - Client ix
- E**
- Error messages 56
 - Evaluation code 16
 - Evaluator copy, installing 16
- F**
- FAMILY, selecting for DATABridge installation 14
 - Files
 - installed 18–22
 - overwritten by installation WFL 11
 - FileXtract
 - creating custom Reader library 49
 - error messages 53
 - handling out-of-order processing 57
 - installation WFL 15
 - installing 14–17
 - overview 2–5
 - using COBOL-to-DASDL utility 37
 - using sample Reader libraries 23
- H**
- Hardware requirements 10
- I**
- Installation
 - access code 16
 - evaluation code 16
 - FAMILY substitution statement 14
 - FileXtract 14–17
 - packs 14
 - results 18–22
 - tape load utility 15
 - usercodes 14
 - WFL for FileXtract 15
 - Italic type, use of vii
 - IVRBICSS (see BICSS Reader library)
- L**
- LINCLOG Reader library
 - activity log files 46–47
 - overview 29
 - selecting 7
 - using 46–47
- M**
- MCP level 10
 - Messages 56
- N**
- NAPFILESDB 26–27
 - Nonusercoded installation 14
- O**
- Out-of-order processing 57
 - Overview
 - COBOL-to-DASDL utility 38
 - custom Reader library 51
 - FileXtract 2–5
 - out-of-order processing 58
 - sample Reader libraries 24
 - using FileXtract 6

P

- Packs, selecting for installation 14
- PRINTFILE Reader library
 - overview 26
 - selecting 6
 - using 32–35
- Processing out-of-order files 57

R

- Reader libraries
 - BANKFILE 7, 30, 32–35
 - BICSS 6, 26–27, 32–35
 - DISKFILE 6, 28, 42–45
 - LINCLOG 29
 - PRINTFILE 6, 26, 32–35
 - sample libraries 24
 - selecting for use 6
 - SUMLOG 6, 25, 32–35
 - TEXT 31, 51–52
 - TTRAIL 6, 25–26, 32–35
 - USERDATA 30
 - using the sample libraries 32–35
- REDEFINES 44
- Replication, related to cloning 62
- Requirements
 - hardware 10
 - software 10

S

- Sample Reader libraries (see also Reader libraries)
 - overview 24
 - using 32–35
- Set
 - definition 64
 - in layout database 2
 - related to DBSupport library 5
- SL (see System library)
- Software requirements 10
- Square brackets, use of vii
- Structure, definition 64
- SUMLOG Reader library
 - overview 25
 - selecting 6
 - using 32–35
- Support, technical 55
- System library, registering DBEngine 17

T

- Technical support 55
- TEXT Reader library
 - overview 31
 - using 51–52
- Tracking
 - definition 64
 - related to cloning 62
- Troubleshooting
 - error messages 56
 - general procedures 54, 55
- TTRAIL Reader library
 - overview 25–26
 - selecting 6
 - using 32–35

U

- Usercode
 - related to FAMILY for installation 14
 - using none for DATABridge installation 14
- USERDATA Reader library
 - overview 30
 - selecting 7
- UVMSBICSS (see BICSS Reader library)

W

- WFLs
 - BANKFILE example 40
 - COBOLTODASDL 44
 - DATABridge installation 15
 - installation 15
 - sample Reader libraries 33

Index
