

Databridge Twin Administrator's Guide

7.1

Table of contents

Databridge Twin Administrator's Guide	4
Abbreviations	4
Conventions	5
Related Documentation	6
Introducing Databridge Twin	7
How Databridge Twin Works	8
Primary and Secondary Database Locations	9
Databridge Twin Components	11
Databridge Limitations	12
Configuring Databridge Twin	13
Configure the Primary Database	13
Configure the Secondary Database	14
Configure Databridge Server for Twin	16
Configure Databridge Twin for the Secondary Database	16
Databridge Twin Parameters	17
Sample Databridge Twin Parameter File	23
Replicating a Database	25
Start Databridge Twin	25
Choose a Cloning Method	26
Tracking	34
Twin Status Report	35
Audit File Numbers Won't Match	35
DASDL Update and Reorganization	36
Handle a DASDL Update	36
Commands Reference	37
Databridge Server Commands (Primary System)	37

Databridge Twin Commands	37
DBTWINCONTROL Data Set	40
About DBTWINCONTROL Data Set	40
General Troubleshooting Procedures	41
Troubleshooting Table	41
Glossary of Terms	44
Legal Notice	48

1. Databridge Twin Administrator's Guide

This guide contains instructions for installing and configuring Micro Focus Databridge Twin. This preface includes information to help you use this guide.

To install, configure, and run Databridge, you should be a system administrator 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)

1.1 Abbreviations

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

Abbreviation	Name
AA	Absolute Address
ABSN	Audit Block Serial Number
AFN	Audit File Number
API	Application Programming Interface
DASDL	Data and Structure Definition Language
DMSII	Data Management System II
IDX	Index
IPC	Inter-Process Communications
MCP	Master Control Program
RPC	Remote Procedure Call
SEG	Segment

Abbreviation	Name
WFL	Work Flow Language

1.2 Conventions

The following conventions and terms may be used in this guide.

This convention or term	Is used to indicate this
<code>this type style</code>	text that you type filenames and directory names onscreen messages
<i>italic</i>	italic variables emphasis document titles
square brackets ([])	optional items in a command For example, [true false]. (Do not type the brackets.)
pipe ()	a choice between items in a command or parameter. When enclosed in braces ({ }), the choice is mandatory.
UPPERCASE	DMSII data set and data item names
MCP server host mainframe (terms)	Unisys ClearPath NX, LX or A Series mainframe
<i>DBEngine</i> (term)	Databridge Engine
<i>DBEnterprise</i> (term)	Databridge Enterprise Server

This convention or term	Is used to indicate this
<i>DBServer</i> (term)	Databridge Server

1.3 Related Documentation

When using Databridge, you may need to consult the following resources.

Databridge product documentation

See [Databridge Documentation](#).

The DOCS folder on the Databridge installation image contains guides for installation, error codes, and the administrator's guides for each Databridge product.

See also the **Help** menu in Databridge Enterprise Server and the Databridge Administrative Console.

Unisys MCP server documentation

If you are not completely familiar with DMSII configuration, refer to your Unisys documentation.

Installing Databridge Twin

See Chapter 7 in the [Databridge Installation Guide](#) for instructions to install Databridge Twin.

2. Introducing Databridge Twin

Databridge Twin is a mainframe program that replicates (clones and then updates) a DMSII database as another DMSII database. The layouts of the two databases must be identical. Their usercodes and families may differ but they must contain exactly the same data sets, sets, and data items.

The original DMSII database is called the primary database, and it is typically a production database. The copy of the primary DMSII database is called the secondary database, and it typically resides on a development or departmental mainframe. Updates to the primary database are applied to the secondary database as DMSII audit becomes available.

- **Primary System**

The primary system is the mainframe that contains the original DMSII database. The primary system must contain the Databridge database replication server software (“Databridge Host”) including DBEngine and DBServer. (For an explanation of the Databridge Accessories, see [accessories](#) or refer to the *Databridge Host Administrator’s Guide*.)

If you have only one mainframe, the primary system is distinguished by the usercode under which the original DMSII database resides.

- **Secondary System**

The secondary system is the mainframe that contains the copy of the DMSII primary database. The secondary system does not require a Databridge Host software installation. It does, however, require the Databridge Twin software. For a list of software components, see [Databridge Twin Components](#) for more information.

If you have only one mainframe, the secondary system is distinguished by the usercode under which the secondary DMSII database resides.

- **Primary Database**

The primary database is always the original DMSII database.

- **Secondary Database**

The secondary database is a duplicate or copy of the primary database except that it may or may not contain the entire contents of the primary database. In all other respects, however, it is a complete DMSII database with its own CONTROL, DESCRIPTION, and audit files.

The secondary database is typically used only for queries and not updates. Some of the query workload can be offloaded from the primary system to the secondary system. Only the Databridge Twin Engine updates the secondary database, and the audit files reflect the updates replicated from the primary database.

2.1 How Databridge Twin Works

Following is a simplified description of the flow of data between the primary and secondary DMSII databases. This description starts after all of the necessary Databridge database replication server software and Databridge Twin components are correctly installed and configured.

1. The Databridge Twin program (secondary system) initiates the Databridge Twin Engine, which calls Databridge Server (primary system).
2. Databridge Server initiates Databridge Engine to retrieve updated information from the audit trail. Databridge Server then sends the updated information to the Databridge Twin Engine.

Note

By default, Databridge Engine processes only closed audit files. However, if you set the Read Active Audit option to true in the Databridge Engine parameter file on the primary system, Databridge Engine can also process the current audit file. To determine how long Databridge Twin waits before checking for available audit files, use the RETRY and MAXWAIT options in the Databridge Twin parameter file. Also note that you can use the DBAuditTimer utility (primary system) to close the current audit file periodically. The DBAuditTimer utility is explained in the [Databridge Host Administrator's Guide](#).

3. When the Databridge Twin Engine receives the audit file data, it does the following by calling routines in the Databridge DMSII Support library:
 - Updates the secondary database (CREATE, LOCK, STORE, and DELETE).
 - Prior to commits, updates either the restart data set or the DBTWINCONTROL data set (whichever is in use) in the secondary database with the current audit location.
2. The Databridge DMSII Support library is a normal application database program. Therefore, all of the usual database operations and exceptions apply to Databridge Twin, such as the possibility for DEADLOCKS, auditing all updates, etc.

The following Databridge Twin features are available:

- REDUNDANT UPDATES, a configuration parameter, specifies whether updates already present in the secondary database should be applied anyway, or discarded.

See the [Sample Databridge Twin Parameter File](#) for details.

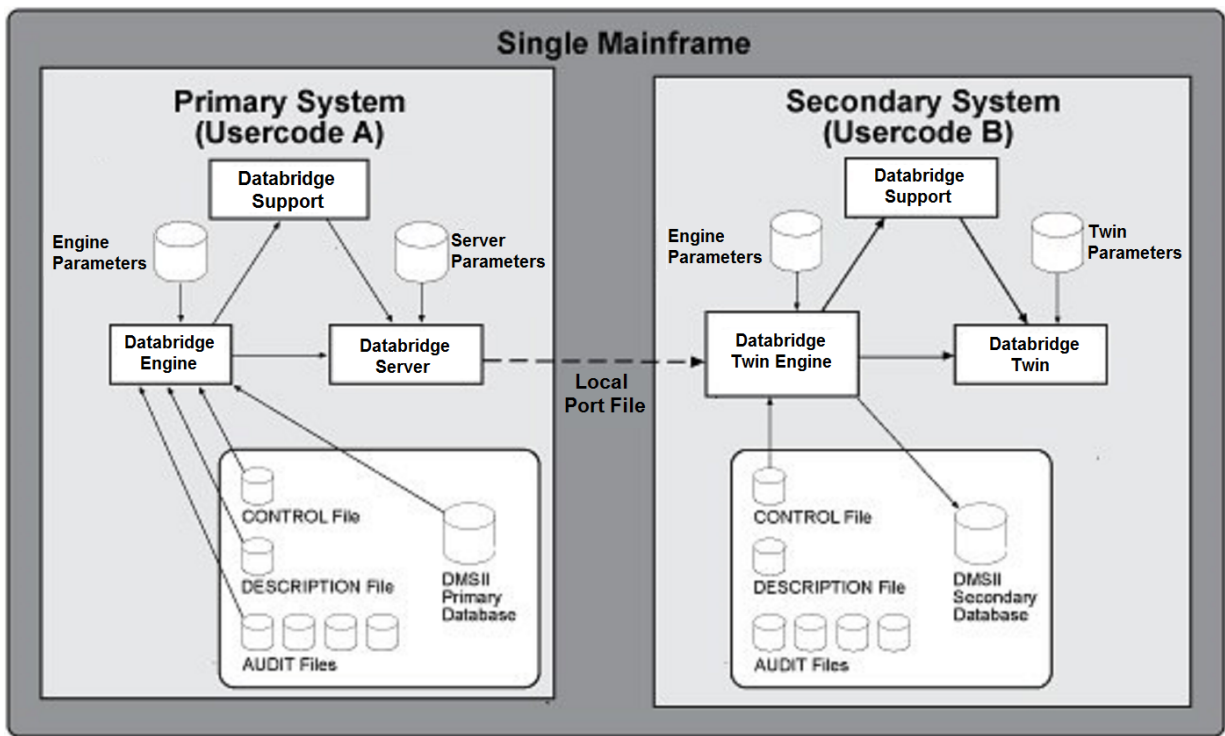
- The tailored support library for the secondary database will apply reformatting specified in an ALTER command. Reformatted data items must be the same size and type as the original item, and virtual items are not allowed.
- A filter in a tailored support library can discard updates for certain records. Filters may not, however, exclude individual data items from a record.
- Databridge Twin writes the audit location and record contents to the report file for any record it cannot find in the client database.
- Databridge Twin can connect to alternate hosts that mirror the primary database or use Remote Database Backup (RDB).
- By default, Databridge Twin writes report files to the DBBD/RUN/TWIN directory on the normal printer backup disk. You can modify WFL/DATABRIDGE/TWIN to change this behavior if desired.

2.2 Primary and Secondary Database Locations

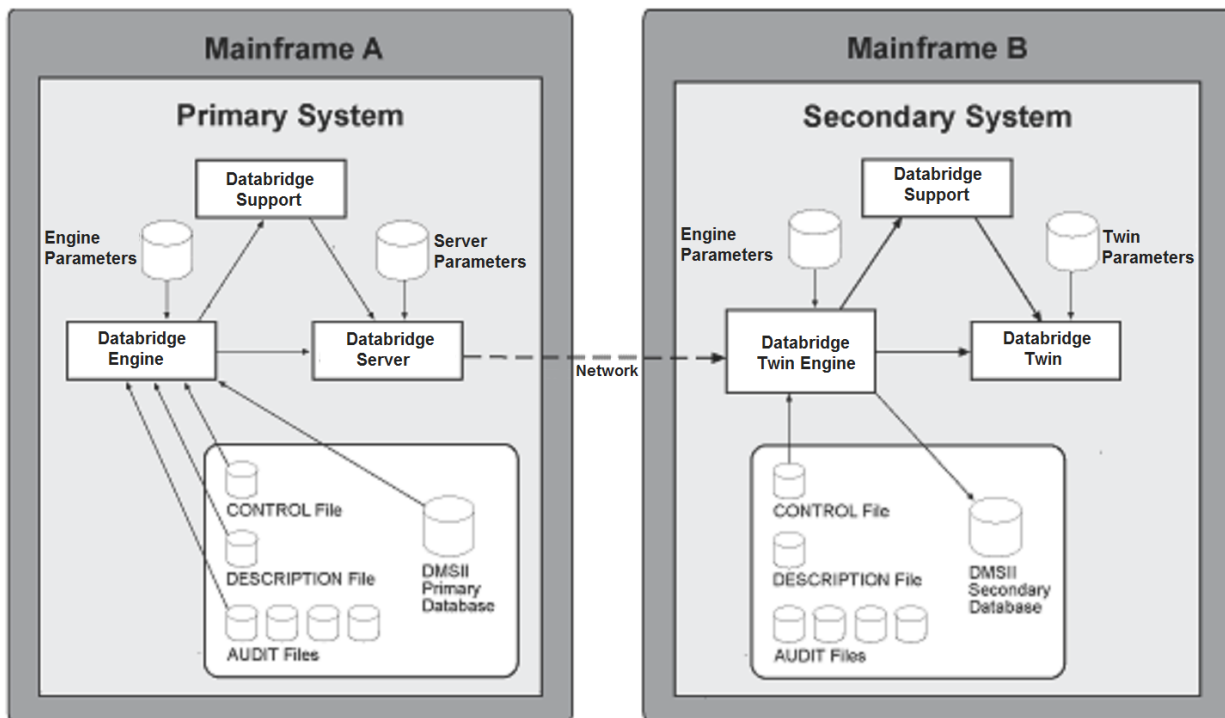
The primary database and the secondary database can reside on the same mainframe or on separate mainframes. If the databases are on the same mainframe, they are differentiated by their usercodes. The primary and secondary systems communicate via a port (network) file using TCP/IP.

The DMSII System Software Release level on the secondary system must match the primary system.

The following diagram shows a typical installation of the Databridge Host and Databridge Twin software on a single mainframe.



The following diagram shows a typical installation of the Databridge Host and Databridge Twin software on separate mainframes.



If you are going to LOAD a database dump from the primary database to create the secondary database and the databases are on separate mainframes, the primary and secondary databases must use the same pack names. Even though the primary and secondary databases physically reside on separate mainframes, the secondary database pack name(s) must be the same as that on the primary system. For example, if the primary database pack is named PRODUCTS, then the secondary database must also reside on a pack named PRODUCTS. This restriction does not apply if you are going to use the CLONE command to create the secondary database.

If the primary and secondary databases are on the same mainframe, you must specify different usercodes in the CONTROL FILE declaration in the DASDL source file.

2.3 Databridge Twin Components

As shown in the previous diagrams, Databridge Twin consists of the following components:

Component	Description
Databridge Server (on the mainframe)	The Databridge Server program calls the Databridge Engine (also on the mainframe), which reads the database CONTROL and audit files on the primary system. When requested, Databridge Server sends the updates to the Databridge Twin Engine running on the secondary system.
Databridge Twin	The Databridge Twin program calls routines in the Databridge Twin Engine to create the secondary database using either a dump from the primary database or by cloning the primary database. From then on, it calls another routine to update the secondary database using the audit information from the primary database.
Databridge Twin Engine	The Databridge Twin Engine is a subset of the full Databridge Engine. The Databridge Twin Engine has the same file name as the full Databridge Engine (OBJECT/ DATABRIDGE/ENGINE). You should not register it as a system library using the SL command. If desired, you can use the full Databridge Engine in place of the special Databridge Twin Engine because it contains all of the same functionality.

Component	Description
Databridge Support Library	The Support Library on the secondary system is the same as the Support Library on the primary system, except that you cannot use column filtering, complex ALTERs, or VIRTUALs. You can use the Support Library to filter the records coming from the primary system so that only certain records are replicated. Note, however, that record filtering is ignored if you use a database dump to create the secondary database. In this case, all records will be loaded into the secondary database. But if you use (record-by-record) cloning to populate the secondary database, the records will be filtered as expected.

2.4 Databridge Limitations

Databridge Twin does not support embedded data sets. Any embedded data sets will be empty in the secondary database.

Databridge Twin does not support RSN (Record Serial Number) data item values. In general, a record's RSN in the secondary database will differ from the same record's RSN in the primary database. Any sets that use the RSN data item as a key item will return a different record in the secondary database than in the primary database. If a data item in a data set contains the RSN of another record it will, in general, refer to a different record in the primary database than in the secondary database.

Databridge Twin will not return an error if a data set has a visible RSN data item, but the results are unpredictable.

3. Configuring Databridge Twin

3.1 Configure the Primary Database

The primary and secondary databases must have identical layouts, i.e. the data sets, sets, data items, etc. must be the same. The values in the PARAMETERS, CONTROL FILE, and AUDIT sections in the DASDL, however, can differ. In fact, if the primary and secondary databases are on the same mainframe the usercode specified in the CONTROL FILE section must be different in the two DASDLs.

If you are going to use a database dump from the primary system to load the secondary database, the pack family names need to stay the same. For example, if the data set PRODUCTS is on DBPACK in the primary database, it needs to be on DBPACK in the secondary database as well. If instead you plan to use cloning to load the secondary database, you can use different family names on the secondary system. (See [Choose a Cloning Method](#) for more information).

Databridge Twin needs to store its state information in the secondary database. It will use the restart data set for this if it has suitable data items. Otherwise, you need to add the DBTWINCONTROL data set to both the primary and secondary databases.

In order to use the restart data set for the Twin state information, it must have one of the following:

- Key or non-key item at least 6 bytes long and a non-key item at least 12 bytes long
- Key or non-key item at least 6 bytes long and two non-key items at least 6 bytes long each
- Non-key item at least 18 bytes long

If your restart data set meets the above requirements, skip to [Configure the Secondary Database](#).

If your restart data set does not meet the above requirements, follow this procedure:

1. Place `PATCH/DATABRIDGE/TWIN/DASDL` into the database DASDL source. The patch defines a data set called DBTWINCONTROL and is explained in [DBTWINCONTROL Data Set](#).
1. Add the following line to the primary and secondary database DASDL:
 1. `$INCLUDE "PATCH/DATABRIDGE/TWIN/DASDL"`
2. If Twin will use a logical database, make sure you enter the data set name DBTWINCONTROL in the data sets list for the logical database if you added the patch.
3. Follow the procedures you normally use at your site prior to performing a DASDL update, such as backing up the `DESCRIPTION/databasename`, `DMSUPPORT/databasename`, and `databasename/CONTROL` files.

Note

This is an update, not a reorganization. You do not have to recompile any other applications that use this database. We recommend that you first do a syntax-only compile to verify that it is a simple update and not a reorganization. The CANDE command would be: `COMPILE AS $databasename WITH DASDL SYNTAX`

4. Do the actual DASDL update compile. The CANDE command would be:
 4. `COMPILE AS $databasename WITH DASDL`
4. This will create a new database DESCRIPTION file.
5. If `$ ZIP` is not set in your DASDL, compile the tailored DMSII software using the following command:
 5. `START DATABASE/WFL/COMPILEACR ("DB=databasename AUDIT=SET")`
5. If `$ DMCONTROL` is not set in your DASDL, update the DMSII CONTROL file using the following command:
 5. `RUN SYSTEM/DMCONTROL ("DB=databasename* UPDATE")`
5. If `$ INITIALIZENEW` is not set in your DASDL and you included the patch for the DBTWINCONTROL data set, initialize it using the following command after you bring down the database:
 5. `RUN $SYSTEM/DMUTILITY ("DB=databasename INITIALIZE DBTWINCONTROL")`
5. After the initialize is complete, you can allow application programs to use the primary database as usual.

3.2 Configure the Secondary Database

1. Copy the DASDL source and DESCRIPTION/ databasename file from the primary system to the usercode and pack of the secondary system.
2. Make the following changes to the DASDL source on the secondary system. You might want to use a patch file for these changes so that they don't have to be repeated if the primary system DASDL changes and you have to recopy it.

Insert `$ RESET DMCONTROL`

Recommended: insert `$ SET ZIP`

Ensure UPDATE (rather than INITIALIZE) is specified

Set the INDEPENDENTTRANS option

Alter the AUDIT TRAIL settings as desired

Update any guardfile titles as necessary

- Put the secondary system usercode in the titles for DMSUPPORT and RECONSTRUCT, and in the CONTROL FILE section

```
CONTROL FILE
(
  USERCODE = secondarydatabaseusercode
);
```

3. SAVE the changes to the DASDL source
4. Compile the secondary database. The CANDE command would be:
4. `COMPILE AS $databasename WITH DASDL`
4. This will create the secondary database DESCRIPTION file.

Note

This is a simple update—not a reorganization. If the DASDL compiler says a reorganization is required there were changes made to the DASDL source that are not permitted. In this case you will need to back out those changes, recopy the primary system DESCRIPTION file and try the compile again.

5. If `$ ZIP` is not set in your DASDL, compile the tailored DMSII software using the following command.
5. `START DATABASE/WFL/COMPILEACR ("DB=databasename AUDIT=SET")`

At this point you will have the DMSII software ready for the secondary database but no database files or CONTROL file. These will be created when you load a dump from the primary database or clone it. See [Replicating a Database](#) for more information.

3.3 Configure Databridge Server for Twin

Use this procedure to configure the Databridge Server software for Databridge Twin.

1. Define a DBServer SOURCE for the primary database in DATA/SERVER/CONTROL for Twin.

1. The SOURCE to which Databridge Twin connects must not include any of the following options:

- 1. NOTIFY
 - AUDIT JOB
 - FILTER
 - STOP
 - PREFILTERED

Warning

The SOURCE must include the following option:

```
TRANSFORM = RAWFORMAT
```

2. Confirm this setting, modify any other part of the file to reflect the settings for your installation, and SAVE the file.

3. If Databridge Server is not already running, enter the following CANDE command to start it:

3. START WFL/DATABRIDGE/SERVER

Refer to **Databridge Server** in the *Databridge Host Administrator's Guide* for more instructions.

3.4 Configure Databridge Twin for the Secondary Database

Use this procedure to configure the Databridge Server software for Databridge Twin. When configuring the Databridge Twin parameter file, note the following:

You can list the options in the parameter file in any order.

You can list multiple options on a single line.

You can split options across multiple lines.

- If you name any entry the same as a parameter file keyword, enclose the name in quotation marks. For example, if you create a filter named SUPPORT (which is also the name of a keyword in the Databridge Twin parameter file), enclose SUPPORT in quotation marks as follows:

```
FILTER "SUPPORT"
```


To configure Twin:

1. On the secondary system, get the Databridge Twin parameter file using CANDE, as follows:
GET DATA/TWIN/SAMPLE/CONTROL AS DATA/TWIN/databasename/CONTROL where databasename is the name of the secondary database. If you are using a logical database, enter the logical database name in place of databasename.
2. Modify the Databridge Twin parameter file (DATA/TWIN/ databasename /CONTROL) to reflect settings for your site. See the following section for a description of the Twin parameters. For a sample configuration file, see [Sample Databridge Twin Parameter File](#).
3. Save the Databridge Twin parameter file.

3.5 Databridge Twin Parameters

Each parameter in the Databridge Twin parameter file is explained as follows.

3.5.1 SOURCE

Required

The SOURCE parameter enables Databridge Twin to link up with Databridge Server. The syntax of the SOURCE parameter is:

```
SOURCE sourcename AT host VIA protocol PORT portnumber [ OR host2 PORT portnumber2 ... ]
```

Where	Is
<i>sourcename</i>	Your entry for the SOURCE option in the Databridge Server parameter file. The SOURCE option in the Databridge Server parameter file is a unique name that is assigned to the primary database.
<i>host</i>	The host name or IP address where Databridge Server is running.
<i>protocol</i>	TCP/IP
<i>portnumber</i>	The Databridge Server TCP/IP port number.
<i>host2</i>	An alternate Databridge Server host to use if the original host becomes unavailable due to a network failure. The alternate host must contain an exact copy of the primary database and audit files as would be the case using Remote Database Backup (RDB) or disk mirroring. Databridge Twin will revert to the original host when it detects that it has more audit available than the alternate host.

Where	Is
<code>portnumber2</code>	The Databridge Server port number of the alternate host.

3.5.2 SUPPORT

Optional

Enter the name of the Support library you want to use for replicating the primary database. The default support library is OBJECT/DATABRIDGE/SUPPORT. If you create a tailored support library for the secondary database, however, enter that file name instead. For an explanation of the Support Library and instructions on creating your own tailored support library, refer to the [Databridge Host Administrator's Guide](#).

Syntax: `SUPPORT title`

3.5.3 FILTER

Optional

Enter the name of the filter you want to use. The filter prevents replication of certain records into the secondary database. (Note that Databridge Twin ignores any column filtering.) You must create and name a filter and compile it in the tailored support library before you can enter a filter here. For instructions, refer to the [Databridge Host Administrator's Guide](#).

Syntax: `FILTER filtername`

3.5.4 RETRY

Optional

The default is 60 seconds. Enter the number of seconds you want Databridge Twin to wait after reaching the end of the available primary database audit. Databridge Twin will wait this many seconds before it tries to read more audit.

Syntax: `RETRY numseconds`

3.5.5 MAXWAIT

Optional

The default is 0 seconds. Enter the maximum number of seconds you want Databridge Twin to wait for more audit to become available. Since Databridge Twin is designed to run continuously, you can enter 0 (zero, which is the default) to indicate there is no limit to the waiting time.

Syntax: `MAXWAIT numseconds`

When you enter a value, Databridge Twin will try to access more audit every *nn* seconds, where *nn* is the value of the `RETRY` option. If the `MAXWAIT` time expires before more audit becomes available, Databridge Twin terminates on the secondary system. Once Databridge Twin terminates, you must restart it manually.

3.5.6 DATABASE TIMEOUT

Optional

The default is 60 seconds. Enter the maximum number of seconds you want Databridge Twin to wait after reaching the end of the available primary database audit on a host before switching to an alternate host, if specified.

Syntax: `DATABASE TIMEOUT numseconds`

3.5.7 STOP

Optional

Use this command when you want Databridge Twin to stop processing when it reaches a certain point in the audit.

You can stop Databridge Twin at a specified quiet point. The quiet point can be before or after a specified time, day, or program. For example, if you wanted to limit daily transactions to only those processed before 4:00 P.M., you would configure the `STOP` command to stop processing at the last quiet point before 4:00 P.M. Once Databridge Twin terminates, you must restart it manually. See [Databridge Twin AX Commands](#) for instructions.

 **Note**

Time in the STOP command refers to the time the update occurred on the primary system, not the current time of day.

The “+/- days” are in relation to the Databridge Twin start date. For example, when Databridge Twin starts, it calculates an Audit location STOP date based on the current date plus or minus the “+/- days” parameter. Databridge Twin stops when it reaches an Audit location from the primary system with this calculated date.

Syntax:

Use one of the following examples of syntax when entering a STOP command:

```
STOP before_or_after timedate STOP [before_or_after "taskname" OR] before_or_after timedate
```

where *before_or_after* is either the word BEFORE or AFTER and *timedate* or "taskname" is a value from the following table. If you specify a *taskname* , you must also specify a *timedate*.

timedate or taskname Value	Sample Entry
<i>hh:mm</i>	<code>STOP BEFORE 22:30</code> This command informs Databridge Twin to stop processing at the last quiet point before 10:30 P.M.
<i>hh:mm AM</i>	<code>STOP AFTER 10:30 AM</code> This command informs Databridge Twin to stop processing at the first quiet point after 10:30 A.M.
<i>hh:mm PM</i>	<code>STOP BEFORE 9:45 PM</code> This command informs Databridge Twin to stop processing at the last quiet point before 9:45 P.M.
<i>hh:mm + days</i>	<code>STOP BEFORE 22:30 + 1</code> This command informs Databridge Twin to stop processing at the last quiet point before 10:30 P.M. tomorrow.
<i>hh:mm - days</i>	<code>STOP AFTER 22:30 - 1</code> This command informs Databridge Twin to stop processing at the first quiet point after 10:30 P.M. yesterday.

timedate or taskname Value	Sample Entry
<i>hh:mm AM + days</i>	<p>STOP BEFORE 10:30 AM + 1</p> <p>This command informs Databridge Twin to stop processing at the last quiet point before 10:30 A.M. tomorrow.</p>
<i>hh:mm PM - days</i>	<p>STOP AFTER 10:30 PM - 1</p> <p>This command informs Databridge Twin to stop processing at the first quiet point after 10:30 P.M. yesterday.</p>
<i>hh:mm ON mm/dd/yyyy</i>	<p>STOP BEFORE 13:30 ON 12/9/2012</p> <p>This command informs Databridge Twin to stop processing at the last quiet point before 1:30 P.M. on December 9, 2012.</p>
<i>hh:mm AM ON mm/dd/yyyy</i>	<p>STOP BEFORE 10:30 AM ON 12/9/2012</p> <p>This command informs Databridge Twin to stop processing at the last quiet point before 10:30 A.M. on December 9, 2012.</p>
<i>hh:mm PM ON mm/dd/yyyy</i>	<p>STOP AFTER 9:30 PM ON 12/9/2012</p> <p>This command informs Databridge Twin to stop processing at the first quiet point after 9:30 P.M. on December 9, 2012.</p>

**timedate or
taskname Value**

Sample Entry

*"taskname" or
timedate*

```
STOP AFTER "OBJECT/SAVINGS/POSTING" OR BEFORE 8:15 PM
```

This command informs Databridge Twin to stop processing at the first quiet point after the EOT of task name "OBJECT/ SAVINGS/POSTING" or at the last quiet point before 8:15 P.M., whichever occurs first.

Keep in mind the following when using the STOP command:

- The STOP command terminates Databridge Twin (it takes Databridge Twin out of the mix). To get Databridge Twin back into the mix, you must manually start Databridge Twin. See [Start Databridge Twin](#).
- If a STOP BEFORE *"taskname"* command is specified, Databridge Twin will stop at the quiet point *before* the task did an OPEN UPDATE on the database. If the task opened the database more than once, Databridge Twin will stop at the last quiet point before the first open.
- If a STOP AFTER *"taskname"* command is specified, Databridge Twin will stop at the quiet point *after* the task closed the database. If the task opened the database more than once, Databridge Twin will stop at the first quiet point after the first close.
- If more than one *"taskname"* is specified, only the last one specified is used. Similarly, if more than one *timedate* is specified, only the last *timedate* specified is used.

3.5.8 FIND

Optional

Normally, Databridge Twin uses the set that does not allow duplicates and contains the fewest number of key elements to find a data set record to update. In situations where a data set has subsets but no sets, you can use this option to tell Databridge Twin which subsets to use to locate records.

Enter the subsets to use when searching for a data set record when there is no set available. Use the following syntax when entering a FIND command:

```
FIND dataset USING subsetlist
```

where *dataset* is the data set Databridge Twin is searching for, and *subsetlist* is the list of subsets you want Databridge Twin to search.

For example, to search the POSTDAY, POSTUSR, and POSTBATCH subsets for a data set called EVENT, you would enter the following command:

```
FIND EVENT USING POSTDAY, POSTUSR, POSTBATCH;
```

When searching for a data set record, Databridge Twin will try the listed subsets in the order specified until it finds the record. List the subsets that most likely contain the data set record first in the subset list. If Databridge Twin does not find the data set record using the listed subsets, it reports an error and the data set record will not be updated.

3.5.9 UPDATERS

Optional

The default is DENIED. Use this parameter to specify if the database should open for exclusive updating by Databridge Twin (recommended) or if it should open to allow other programs to update the database concurrently with Databridge Twin.

Syntax:

```
UPDATERS [ DENIED | ALLOWED ]
```

DENIED - Only Databridge Twin can update the database.

ALLOWED - One or more programs can update the database at the same time as Databridge Twin.

3.6 Sample Databridge Twin Parameter File

Databridge Twin uses a SEQDATA file for its parameter file (DATA/TWIN/ databasename / CONTROL). This file provides information to the Databridge Twin program on where and how to locate the Databridge Server SOURCE of the primary database.

Each parameter in this file is explained in the [Databridge Twin Parameters](#) section.

```

%-----
%
% Copyright 2019 Micro Focus or one of its affiliates.
%
% Module: DATA/TWIN/SAMPLE/CONTROL
%
% Project: Databridge
%
% Description: Databridge Twin Parameter File
%
% Copyright 2019 Micro Focus or one of its affiliates.
%
%-----

% How to locate DBServer SOURCE ...

SOURCE <sourcename> % SOURCE name in DBServer parameter file
  AT <host> % DBServer's hostname or IP address
  VIA <protocol> % network protocol, e.g. TCPIP
  PORT <portnumber> % DBServer's port number, e.g. 11367

% example: SOURCE BANKDB AT PRODHOST VIA TCPIP PORT 11367

% Filter in DBSupport selects which records ...

SUPPORT OBJECT/DATABRIDGE/SUPPORT % title of DBSupport
FILTER DBFILTER % filter entrypoint name in "

% When waiting for an audit file ...

RETRY 60 % seconds delay between retries
MAXWAIT 0 % max total seconds to wait

% (0 means 'forever')

% When to stop processing ...

% STOP AFTER "<program name>" OR BEFORE hh:mm PM ON mm/dd/yyyy

% How to find records using subsets if no sets

% FIND dataset USING subset1, subset2, ...

% Allow other update programs ...

UPDATERS DENIED % DENIED - no other updaters (default)
                % ALLOWED - other programs can update
                % the client database while Twin
                % is running

```


4. Replicating a Database

This chapter discusses the replication process, which involves two processes: performing the initial clone and then tracking updates to the database.

4.1 Start Databridge Twin

Use the following procedure to start Databridge Twin.

To start Databridge Twin

- Use the following command:

```
START WFL/DATABRIDGE/TWIN ("secondarydatabasename" [ , ["logicaldatabasename" ] ,  
"taskstring" ])
```

Where	Is
<i>secondarydatabasename</i>	The title of the secondary DMSII database DESCRIPTION file without the "DESCRIPTION" node.
<i>logicaldatabasename</i>	The name of the DMSII logical database you are replicating. If you are not replicating a logical database, omit this parameter but include the comma.
<i>taskstring</i>	One of the following commands: STATUS - Prints a report of current audit locations as indicated by the DBTWINCONTROL or restart data set LOAD <i>dumplist</i> – loads the secondary database using dumps from the primary database CLONE – loads the secondary database by reading each record in the primary database LOCATION <i>afn, absn, seg, inx</i> – stores the specified audit location in the secondary database as the starting point for tracking when the secondary database has been loaded using means other than the LOAD or CLONE command UPDATE

When you run Databridge Twin with the UPDATE command, it runs continuously until it is interrupted by a STOP command or a MAXWAIT timeout or it encounters an unhandled DMSII exception.

4.2 Choose a Cloning Method

Databridge Twin provides three ways to perform an initial clone. After you determine the appropriate method for your site, proceed to the procedure for that method.

Before you can clone, you must have completed installation and configuration on the primary and secondary systems.

Use this method	To
CLONE command	Clone the primary database, record by record. This method is similar to how the Databridge Client clones. Choose this method if you have a filter that excludes records from the secondary database.
LOAD command	Dump the primary database and then load it to the secondary database. Choose this method if you do not have a filter and want the fastest clone time for a large database.

Use this method	To
LOCATION command	Populate the secondary database with user-written programs. Choose this method if the CLONE and LOAD methods are unsuitable for some reason. The LOCATION command gives Databridge Twin the audit location in the primary database so that Databridge Twin can run in its normal tracking mode.

Keep in mind that the amount of time required to populate the secondary database depends on several factors, including the size of the DMSII database you are replicating and the type of mainframes you are using.

4.2.1 Clone Using the CLONE Command

Use this procedure to perform the initial clone using the CLONE command.


If you previously cloned the primary database using a method other than the Databridge Twin CLONE or LOAD commands, skip this procedure and run Databridge Twin with the [LOCATION command](#).

To clone

1. On the secondary system, sign on to the usercode where the Twin software was installed.
2. Start Databridge Twin by entering the following:
 2. `START WFL/DATABRIDGE/TWIN ("secondarydatabasename", "logicaldatabasename", "CLONE")`

Parameter	Description
<i>secondarydatabasename</i>	Name of the DMSII database on the secondary computer. If you also specify a usercode, make sure it is the usercode of the secondary database DESCRIPTION file.
<i>logicaldatabasename</i>	Optional. Name of the DMSII logical database. This is the logical database you want to clone into the secondary database. If you are not using a logical database, just put a second comma after the first parameter.
CLONE	Required literal.

2. Databridge Twin resets any family name overrides found in the database CONTROL file and updates the database CONTROL file. Then it requests a clone of the datasets via Databridge Server and Databridge Engine on the primary system, which sends the requested records.

 **Note**

If the Databridge Twin clone fails for any reason, remove all of the secondary database files, including the database CONTROL file and the audit files, and go back to step 2. (You can leave the DESCRIPTION, DASDL, DMSUPPORT, and RECONSTRUCT files.)

2. When the clone is complete, Databridge Twin prints a status report with a title such as:
 2. `(DB70)DBBD/RUN/TWIN/BANKDB/0002241/0002242/000000000000/STATUS`

The following is a sample report for a database called TESTDB:

DBTwin Version 6.6.0.001 compiled Wednesday, April 19, 2017 @ 05:12:28

Source TESTDBTWIN at
MARS port 7000
172.27.172.91 port 7000

Filter DBFILTER in OBJECT/DATABRIDGE/SUPPORT
Retry audit every 6 seconds for at most 10 seconds
Database timeout 60 seconds
Other update programs not allowed

DBTwin Status Report for TESTDB as of Thursday, December 3, 2015 @ 17:33:42

Str#	Name	Mode	AFN	ABSN	Segment	Index	Host
	(common)	clone					
4	MASTERDS	clone					
8	D	normal	6	68	1	10	BB4
10	COUNTRIES	normal	6	68	1	10	BB6
16	BACKORDS	normal	6	68	1	10	BB5
18	EMPLOYEE	normal	6	68	1	10	BB8
20	CUSTOMER	normal	6	68	1	10	BB4
22	ORDDTAIL	normal	6	68	1	10	BB5
24	PRODUCTS	normal	6	68	1	10	BB8
26	SUPPLIER	normal	6	68	1	10	BB4
28	ORDERS	normal	6	68	1	10	BB5
30	JUAN	normal	6	68	1	10	BB6
32	DIR	normal	6	68	1	10	BB5
34	CANDO	clone					

Updates Create:0 Modify:0 Delete:0
Errors Create:0 Modify:0 Delete:0

Datasets that have a different audit location or a mode other than normal will be printed on a separate line in the report.

Databridge Twin then switches to the UPDATE command and stays in the mix, replicating any additional updates that occurred on the primary database during and after the clone. The secondary database (named the same as the primary database) is now available for queries. Databridge Twin runs continuously until interrupted (for example, by a STOP command, MAXWAIT parameter setting, or unhandled DMSII exception).

4.2.2 Clone Using the LOAD Command

Use the LOAD command with a dump file as an alternative to cloning using the CLONE command. This method is performed in two separate procedures. The first procedure occurs on the primary system; the second occurs on the secondary system.

Note

This method bypasses any filters you specified in the Databridge Twin parameter file. In other words, once the dump file has been loaded, the secondary database will have all of the records that were dumped from the primary database, not just those that satisfy the filter.

Start the Clone on the Primary System

Use this procedure first when performing an initial clone using a LOAD of a dump file. Complete the process by following the instructions in [Finish the Clone on the Secondary System](#).

To start the initial clone

1. Dump the primary database to tape or to disk:

To	Do this
Dump to tape	Dump the entire database to tape using the following CANDE command: <pre data-bbox="437 389 1289 416">RUN \$SYSTEM/DMUTILITY ("DB=databasename DUMP = TO dumpname")</pre> -or- Clone a logical database by dumping (at least) the data sets and sets that belong to the logical database. For example, if the logical database contains data sets D1 and D2 in a physical database called MYDB, you would use the following CANDE command: <pre data-bbox="437 790 1305 864">RUN \$SYSTEM/DMUTILITY ("DB=MYDB DUMP MYDB/D1/=,MYDB/D2/= TO MYDBDUMP")</pre>

To	Do this
Dump to disk	<p>Dump the entire database to a disk file under the primary database usercode by entering:</p> <pre>RUN \$SYSTEM/DMUTILITY ("DB=databasename DUMP = TO dumpname ON packname")</pre> <p>-OR-</p> <p>Dump selected data sets to the primary database usercode by entering:</p> <pre>RUN \$SYSTEM/DMUTILITY ("DB=databasename DUMP datasetdirectories TO dumpname ON packname")</pre> <p>Then, change the security of the dump file on the disk to PUBLIC IN so that the secondary usercode can read it, as in the following example:</p> <pre>SEC dumpname ON packname PUBLIC IN</pre>

2. You can now use the primary database as normal.

Note

Run at least one program that does a database OPEN UPDATE before proceeding.

3. Preserve the audit files covering the period from the start of the dump to the end of the dump for use on the secondary system.
4. If you wrote the dump to tape, move the tape(s) to the secondary system. If you wrote the dump to disk instead of tape, use BNA or whatever is appropriate at your site to copy the dump to the secondary system.
4. At this point, you have the primary database dump (on tape or disk). Now you are ready to start the cloning process on the secondary system, as described in the next section.

Finish the Clone on the Secondary System

Use this procedure to complete the initial cloning process started in [Start the Clone on the Primary System](#). In this procedure, you'll create the cloned database using the LOAD command.

To complete the initial clone

1. Do one of the following:

If you dumped the primary database to tape, mount the tape on the secondary system.

If you dumped the primary database to disk, make sure that you copy the dump file(s) to the secondary system.

2. Sign on to the usercode containing the Databridge Twin software and enter the following to start Databridge Twin:

2. `START WFL/DATABRIDGE/TWIN`

2. (`" secondarydatabasename ", " logicaldatabasename ", "LOAD dumpname [ON familyname]"`)

Where	Is
<i>secondarydatabasename</i>	The name of the DMSII database on the secondary system. Note: If you specify a usercode, make sure it is the usercode of the secondary database DESCRIPTION file.
<i>logicaldatabasename</i>	The name of the DMSII logical database you want to clone on the secondary system. If you are not using a logical database, omit this parameter but include the comma as shown in the following example: <code>("secondarydatabasename" , , "LOAD dumpname")</code>
dumpname –or–	The name of the dump(s) from the primary database. If you used the example in Start the Clone on the Primary System , the dump name is <code>MYDBDUMP</code> .
dumpname, dumpname, ...	You can also specify a comma-delimited list of tape dump names for the dump parameter. This allows partial dumps to be combined into a complete secondary database. The following is an example of a comma-delimited list of tape dumps: <code>START WFL/DATABRIDGE/TWIN("BANKDB" , , "LOAD MONDAYDUMP, THURSDAYDUMP, CUSTDUMP")</code> The default is tape dumps. If you dumped to disk, be certain to include ON familyname in the dump name.

Where	Is
LOAD	A required literal that initializes the secondary database and loads the structures from the dumps taken on the primary system.

2. Databridge Twin resets any family name overrides in the database CONTROL file, updates the database CONTROL file, and then initiates DMUTILITY to load the dump. In addition, Databridge Twin will request one or more audit files by displaying a NO FILE message on the terminal and ODT. The messages look similar to the following:

2. NO FILE *databasename* /AUDIT *nnnn* (MT) #1

3. Copy the requested audit file from the primary system to the secondary system and usercode.

3. If you use sectioned audit files, be sure to copy all of the sections. Then give the waiting task an FA command:

3. *mixnumber* FA TITLE = *databasename*/ AUDIT *nnnn* ON *family*

4. Watch for additional requests for audit files and repeat the previous step if necessary.

4. If the requested audit file is in use on the primary system because it is the current audit file, either wait for DMSII to switch to the next audit file or force a switch immediately. To force an audit file switch, find the mix number of the database by entering the DBS command from the ODT or MARC, and then enter *mixnumber* SM AUDIT CLOSE FORCE.

Note

If Databridge Twin fails for any reason, remove all of the secondary database files, including the database CONTROL file and the audit files. (Leave the DESCRIPTION, DASDL, DMSUPPORT, and RECONSTRUCT files.) Then, go back to step 2.

4. After the load, Databridge Twin prints a status report. See [Twin Status Report](#) for more information.

4. Then, Databridge Twin stays in the mix (automatically switching to the UPDATE command), replicating any additional updates that occurred on the primary database since the end of the dump. The secondary database (named the same as the primary database) is now available for queries. Databridge Twin runs continuously until interrupted (for example, by a STOP command, MAXWAIT parameter setting, or unhandled DMSII exception). For information on manually starting Databridge Twin, see [Tracking](#) for more information.

4.2.3 Clone Using the LOCATION Command

If you cloned the primary database using a method other than the Databridge Twin LOAD and CLONE commands, you must specify the audit file location. The LOCATION command provides the audit location in the cloned database so that Databridge Twin can run in its normal tracking mode.

To specify the audit file location

- Run Databridge Twin with the LOCATION command, as follows:

```
START WFL/DATABRIDGE/TWIN ( "secondarydatabasename" [ , "logicaldatabasename" ] ,  
"LOCATION afn absn seg inx" ] )
```

Databridge Twin stores this audit location in the secondary database as the starting point for tracking. Databridge does not validate the specified audit location until the next time Databridge Twin is run in tracking mode.

Databridge Twin prints a STATUS report after storing the new location.

Note

Incorrect use of the LOCATION command can cause the client database to miss the updates that would have occurred before the specified audit location.

When you're done, use the primary database as you usually would, using the secondary database only for queries.

4.3 Tracking

After you have performed the initial clone, Databridge Twin automatically retrieves changes from the audit files and applies them to the replicated database.

If you have set the STOP command to stop Databridge Twin, or if you have set the MAXWAIT parameter to something other than 0, Databridge Twin will stop when the STOP or MAXWAIT conditions are met. In this case, you must restart Databridge Twin using the following command:

```
START WFL/DATABRIDGE/TWIN ( "secondarydatabasename" [ , "logicaldatabasename" ] )
```

If you find that data in the secondary database is not current enough, you can also set the Read Active Audit option to true in the Databridge Engine parameter file on the primary system for near real-time access to the audit trail. Alternatively, you might consider adjusting the number of times audit files close on the primary database. You can accomplish this via the DBAuditTimer utility, which is described in the [Databridge Host Administrator's Guide](#).

⚠ Caution

Programs that update the same records in the secondary database as Databridge Twin can cause Databridge Twin to encounter various DMSII errors such as NOTFOUND or DUPLICATES. To prevent these problems, consider using guardfiles, which are explained in the Unisys mainframe documentation. Some update programs, such as LINC-generated report programs, can generally coexist with Databridge Twin because they don't actually update the same records as Databridge Twin.

4.4 Twin Status Report

Databridge Twin writes a report each time it runs showing various parameter values and the current audit location. The internal name of the report is REPORT. By default, it is a printer file whose external name is STATUS. The title of the report is *(secondaryusercode)DBBD/RUN/TWIN/databasename/jobnumber/tasknumber/000000000000/STATUS*

If a dataset has a mode other than normal or a different audit location from the others, it will be printed on a separate line in the report.

Here is a sample report:

```
DBTwin Version 6.6.0.001 compiled Tuesday, April 19, 2017 @ 05:12:28

Source TESTDBTWIN at
  MARS port 7000
  172.27.172.91 port 7000

Filter DBFILTER in OBJECT/DATABRIDGE/SUPPORT
Retry audit every 6 seconds for at most 10 seconds
Database timeout 60 seconds
Other update programs not allowed

DBTwin Status Report for TESTDB as of Thursday, December 3, 2015 @ 17:33:42

Str#  Name                Mode  AFN  ABSN  Segment  Index  Host  info
-----
4     (common)                clone
4     MASTERDS                clone
8     D                        normal 6    68    1        10     BB4
10    COUNTRIES               normal 6    68    1        10     BB6
16    BACKORDS                normal 6    68    1        10     BB5
18    EMPLOYEE                normal 6    68    1        10     BB8
20    CUSTOMER                normal 6    68    1        10     BB4
22    ORDDTAIL                normal 6    68    1        10     BB5
24    PRODUCTS                normal 6    68    1        10     BB8
26    SUPPLIER                normal 6    68    1        10     BB4
28    ORDERS                  normal 6    68    1        10     BB5
30    JUAN                    normal 6    68    1        10     BB6
32    DIR                     normal 6    68    1        10     BB5
34    CANDO                  clone

Updates Create:0 Modify:0 Delete:0
Errors Create:0 Modify:0 Delete:0
```

4.5 Audit File Numbers Won't Match

In most cases, the audit file numbers on the secondary system will not match those on the primary system after tracking begins. Even though both databases start with the same audit file number, the secondary database may close its audit files more or less frequently than the primary database. Remember that the only program updating the secondary database should be Databridge Twin. All other secondary database activity should be from queries only.

5. DASDL Update and Reorganization

This chapter describes how to use Databridge Twin to handle a DASDL update/reorganization of the primary database. Garbage collection reorganizations on the primary database do not impact the secondary database and can be ignored.

5.1 Handle a DASDL Update

Databridge Twin requires the descriptions of the primary and secondary database structures to be identical and the update timestamps of the DESCRIPTION files to be identical. A DASDL update generates a new update level and update timestamp. Twin cannot process audit files written by the new update level from the primary system if the secondary database is at the old update level. The secondary database cannot be reorganized to match the primary database because the update timestamps will not match even though they would be at the same update level.

For this reason when the DASDL is updated on the primary system, the secondary system will have to be generated and cloned from scratch.

Use this procedure with Databridge Twin to manage updates and reorganizations to the primary database.

To update a DASDL

Remove the secondary database files.

Perform all of the steps in the “Configure the secondary database” section.

Clone the primary database using your preferred method.

6. Commands Reference

This chapter explains the Databridge Server commands and the Databridge Twin commands.

6.1 Databridge Server Commands (Primary System)

Note that Databridge Server has a parameter file that you must configure. Refer to the *Databridge Host Administrator's Guide* for information about configuring the DBServer parameter file.

6.1.1 Start Databridge Server

To start Databridge Server

- Enter the following from CANDE or MARC under the usercode containing the Databridge software:

```
START WFL/DATABRIDGE/SERVER
```

The WFL starts Databridge Server, which runs continuously.

6.1.2 Terminate Databridge Server

In most cases, you do not need to terminate Databridge Server.

To terminate Databridge Server

- Enter the following from CANDE, MARC, or the ODT under the usercode containing the Databridge software:

```
mixnumber AX QUIT
```

where *mixnumber* is the task number of OBJECT/DATABRIDGE/SERVER.

6.2 Databridge Twin Commands

Databridge Twin includes the following commands, which are only applicable on the secondary system.

| Use this command | To do this | | CLONE | Replicate the primary database from the secondary system usercode that will contain the secondary database. For instructions, including syntax, see [Clone Using the CLONE Command](#). | | LOAD | Replicate the primary database. For instructions, including syntax, see [Clone Using the LOAD Command](#). | | LOCATION | Give Databridge Twin the audit location in the cloned database so that Databridge Twin can run in its normal tracking mode. (For the LOCATION command syntax, see [Clone Using the LOCATION Command](#).) | | STATUS | Print a report of current audit file locations as indicated by the DBTWINCONTROL or restart data set. For syntax, see [Start Databridge Twin](#). | | UPDATE or TRACK or empty | Cause Databridge Twin to perform normal tracking. Changes are then retrieved from the audit file and applied to the replicated database. For syntax, see [Start Databridge Twin](#). |

6.2.1 Databridge Twin AX Commands

The following AX commands are available for Databridge Twin. Enter them on the secondary system.

Command	Description
<code>mixnumber AX</code> <code>QUIT</code>	Terminates Databridge Twin at the end of the next transaction group.

Command	Description
<code>mixnumber AX STATUS</code>	Displays Databridge Twin's location and timestamp in the primary database audit file, as well as the number of records replicated to the secondary database. See the following sample output.

AX STATUS

Use AX STATUS to view the current audit location and record counts. The line starting with Updates as the first word is the number of successful updates of each type. The line starting with Errors is the number of failed updates of each type.

Syntax:

```
mixnumber AX STATUS
```

where *mixnumber* is the task number of OBJECT/DATABRIDGE/TWIN

Sample Output for Databridge Twin AX STATUS Command

```
DBTwinEngine: Waiting for AFN 41 to become available.
DBTwinEngine: Errors Create:0 Modify:0 Delete:0.
DBTwinEngine: Updates Create:0 Modify:3 Delete:0.
DBTwinEngine: Audit time: month day, year @ 16:42:52.
DBTwinEngine: Audit location: AFN=40 ABSN=3552 (70%).
```

In this example, (70%) means that 70 percent of audit file number (AFN) 40 has been processed.

AX QUIT

Use AX QUIT to stop Databridge Twin. In most cases, terminating Databridge Twin is not necessary.

Syntax:

```
mixnumber AX QUIT
```

where *mixnumber* is the task number of OBJECT/DATABRIDGE/TWIN.

7. DBTWINCONTROL Data Set

The information in this section explains the DBTWINCONTROL data set as mentioned in [Configure the Primary Database](#).

7.1 About DBTWINCONTROL Data Set

DBTWINCONTROL contains audit locations that are maintained on a structure-by-structure basis. The DBTWINCONTROL data set will reside in both the primary and the secondary databases, although the DBTWINCONTROL data set is always empty in the primary database. In the secondary database, it can contain a record for each data set you clone from the primary database to the secondary database.

DBTWINCONTROL keeps track of the respective audit locations for each data set cloned from the primary database. Typically, one global record keeps track of all of the data sets that are in normal mode (indicated by the number 2 in DBTWIN-MODE) and up-to-date.

Data sets that are not in normal mode have their own record that reflects their mode (for example, 3 for reorganization or 4 for purged). This situation is rare, however. The majority of the time, DBTWINCONTROL will have only the global record. (For an explanation of normal and other modes, see the [Datbridge Host Administrator's Guide](#).)

7.1.1 DASDL source code for the DBTWINCONTROL data set

```
% DATBRIDGE TWIN ADDITIONS
```

```
DBTWINCONTROL DATASET ( DBTWIN-STRNUM REAL (S11) INITIALVALUE 0; DBTWIN-RECTYPE  
REAL (S11) INITIALVALUE 0; DBTWIN-AFN REAL (S11) INITIALVALUE 0; DBTWIN-ABSN REAL (S11)  
INITIALVALUE 0; DBTWIN-SEG REAL (S11) INITIALVALUE 0; DBTWIN-INX REAL (S11) INITIALVALUE  
0; DBTWIN-TIME REAL INITIALVALUE 0; DBTWIN-MODE REAL (S11) INITIALVALUE 0; DBTWIN-  
FORMAT-LVL REAL (S11) INITIALVALUE 0; DBTWIN-TABLE-LVL REAL (S11) INITIALVALUE 0;  
DBTWIN-ITEM-COUNT REAL (S11) INITIALVALUE 0; DBTWIN-OPTIONS REAL (S11) INITIALVALUE 0;  
% added 6.5 DBTWIN-HOST-INFO REAL (S11) INITIALVALUE 0; % added 6.5 ), BLOCKSIZE = 22  
SEGMENTS; % for VSS-3 compatibility
```

```
DBTWINSET SET OF DBTWINCONTROL KEY (DBTWIN-STRNUM, DBTWIN-RECTYPE) NO  
DUPLICATES; ``
```


8. General Troubleshooting Procedures

If you have problems using Databridge Twin, complete the following steps:

1. Make sure that your system meets the requirements necessary to use the product. See the [Databridge Installation Guide](#) for system requirements.

2. If your primary and secondary systems are on separate mainframes, ensure that the mainframes are communicating, as follows:

Try the PING function.

3. Check your setup, as follows:

Is Databridge Server running on the primary system? Check the mix to make sure. If it is not running, see [Start Databridge Server](#) for more information.

If your restart data set did not meet the criteria, did you use the INCLUDE statement or copy the DBTWINCONTROL patch to your DASDL source?

4. Check your configuration options for connecting to the mainframe and the DMSII primary database, as follows:

Verify that SOURCE in the Databridge Twin parameter file is pointed to the correct Databridge Server and SOURCE on the primary system. For details, see [SOURCE](#) for more information. Also check the DBServer parameter file and make sure the entry for SOURCE is correct.

Verify that your entries in the Databridge Twin parameter file match your entries in the DBServer parameter file, especially the port number.

5. Check the status of the Databridge Twin Engine. For instructions, see [Databridge Twin AX Commands](#).

6. Resolve any errors that you receive. If you receive error messages or status messages that you don't understand, see the [Databridge Error and Messages Guide](#) on the installation image for help resolving these messages.

7. 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.

8. Troubleshoot the problem using information available from the [Knowledge Base](#) articles.

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

9. Contact [Customer Care](#).

8.1 Troubleshooting Table

The following table lists common problems and their solutions:

Problem	Resolution
Databridge Twin gets a PORT file I/O error and terminates	Start Databridge Server on the primary system. Server must be running for Databridge Twin to contact it. If Server is not running, Databridge Twin tries until it eventually gets a PORT file I/O error and terminates.
Any of the following occur: Disk crash on the secondary system. Any reorganization of the primary database. Loss of communications between the primary and secondary systems for enough time that it is less expensive to recreate the secondary database than it is to process all of the audit files. For example, if the primary database is relatively small but highly volatile, it would most likely be less expensive to recreate the secondary database.	Reclone the secondary database on the secondary system by removing all of the database and audit files and then repeat the replication procedure as described in Start the Clone on the Primary System and Finish the Clone on the Secondary System .

Problem

A message appears stating that the DBTWINCONTROL data set is not defined or is not available.

Resolution

Check the following:

Did you compile the DASDL after adding the \$INCLUDE "PATCH/DATABRIDGE/TWIN/DASDL" statement or after inserting the contents of PATCH/DATABRIDGE/TWIN/DASDL?

Did you initialize DBTWINCONTROL using DMUTILITY? For instructions, see Chapter 2, [Configuring Databridge Twin](#).

9. Glossary of Terms

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.
- 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.
- DBInfo, which produces a report of your DMSII database timestamps, update 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.
- DBAuditTimer, which closes the current audit file when it is older than a specified length of time.

Accessroutines

The Accessroutines program is a DMSII library program that controls access to the database, reads and writes records, and creates the audit trail.

audit file

An audit file is created by the DMSII Accessroutines and contains the raw format of changes made to the DMSII database by update programs. Audit file records contain the deletes, creates, and modifies that were made to the various structures. Depending on the frequency of changes made to a database, the information in an audit file can span a few hours or several weeks.

Databridge uses the audit file for the raw data of each database change to exactly replicate the primary database. Databridge records the audit location (AFN, ABSN, SEG, IDX) between runs, so it can restart without losing any records.

audit trail

The audit trail contains all of the audit files generated for a database. The Databridge Engine reads the audit files to extract updates. It then passes the updates to the Client to be applied to the relational database. After the updates have been successfully extracted, the Client saves the state information, which includes the location in the audit trail from which the last group of updates for the data set were read.

cloning

Cloning is the one-time 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 synchronized 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 (or updating), which is the integration of database changes since the cloning.

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/databasename).

Databridge uses the CONTROL file for the following information:

- Timestamps

- INDEPENDENTTRANS option

- AFN for the current audit file and ABSN for the current audit block

- Data set pack names

- Audit file pack name

- Database user code

DASDL

Data and Structure Definition Language (DASDL) is the language that defines DMSII databases. The DASDL must be compiled to create a DESCRIPTION file.

data set

A data set is 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.

Databridge Engine

The Databridge Engine (also referred to as DBEngine) is a host library program that uses the DMSII Support Library to retrieve data records from the DMSII database for cloning.

DESCRIPTION file

The DESCRIPTION file contains the structural characteristics of a database, physically and logically. This file is created from the DASDL source by the DASDL compiler and contains the layout (physical description), timestamp, audit file size, update level, logical database definition, and any static information about the database. It contains information about the database, not the data itself.

There is only one current DESCRIPTION file for each DMSII database. Databridge must have access to the DESCRIPTION file before it can replicate a database. Additionally, Databridge uses the DESCRIPTION file information for consistency checks between the primary database and the secondary or replicated database.

The DESCRIPTION file corresponds to the schema in a relational database.

DMSII Support

DMSII Support is a Databridge library that retrieves data records from the DMSII database for cloning. The Databridge Engine links to this library to perform database functions such as reading records, switching the audit file, and getting database statistics.

extraction

Extraction is the process of reading through a data set sequentially and writing those records to a file (either a secondary database or flat file).

fixup records

Changes that occur to the DMSII database while a clone is taking place.

primary database

This is the original DMSII database that resides on the host. Databridge replicates from the primary database to one or more client databases. The client databases can be another DMSII database or one of several relational databases. Compare this to the replicated (or secondary) database.

quiet point (QPT)

A quiet point is a point in the audit trail when the DMSII database is quiet and no program is in transaction state. This can occur naturally, or it can be forced by a DMSII sync point.

replication

Replication is the ongoing process of cloning and tracking changes to a DMSII database.

secondary database

The replicated database. The replicated database is the database that usually resides on the client machine and contains records cloned from the DMSII database. The replicated database is updated periodically with changes made to the primary (original) DMSII database. The periodic update (or tracking process) is explained later in this section. Compare this to the primary database.

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.

subset

An index into a data set. A subset does not necessarily have an entry (key + pointer) for every record in the data set. Subsets are used to access selected members of a data set and to represent relationships between data set records. Subsets typically contain fewer entries than normal sets.

An automatic subset is any subset that contains a WHERE clause and is maintained by DMSII.

A manual subset is any subset that is maintained by an application.

Support Library

A library that provides translation, formatting, and filtering to the DBServer and other accessories. After DBServer receives data from the Databridge Engine, it calls the Support Library to determine if the data should be replicated, and if so, passes the data to the Support Library for formatting.

tracking

Tracking is an ongoing process for propagating changes made to records in the DMSII primary database to the replicated database after the initial clone. The Databridge Engine performs extraction as well as tracking.

10. Legal Notice

Copyright 2023 Open Text

The only warranties for products and services of Open Text and its affiliates and licensors ("Open Text") are as may be set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Open Text shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice. .