Flat File Client Administrator's Guide

7.0

None

Table of contents

1. About This Guide

This guide contains instructions for configuring and using Micro Focus Databridge Flat File Client. This preface includes information to help you use this guide.

This guide complements the information found in the Databridge Client Administrator's Guide. Most of the information in that guide also applies to the Flat File Client, so rather than repeat the information in both guides, we will often refer you to the Databridge Client Administrator's Guide for information.

You can use this guide with either the command-line or service-controlled Flat File Client. For information specific to the service-controlled Client, see the following topics in the Databridge Client Administrator's Guide or go to the Help menu in the Databridge Client Console and search for these topics:

- Chapter 1: Using the Client Console and the Service
- Chapter 8: Automating Client Operations with the Service

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)
- File layouts and the description of those layouts for the files you will be replicating

1.1 Conventions

We may use any of the following conventions and terms in this guide:

+------------------+--+ | **This convention** | **Is used to indicate this** | | **or term** | | +==================+==+ | `this type style` | text that you type | | | | | | filenames and directory names | | | | | | on-screen messages | +------------------+--+ | *italic* | document titlesvariablesemphasis | +------------------+--+ | 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*, | Unisys ClearPath NX, LX or A Series mainframe | | *host*, | | | *mainframe*, | | | (interchangeable | | | terms) | | +------------------+--+ | *DBEnterprise* | Databridge Enterprise Server | | (term) | | +------------------+--+ | *DBServer* (term) | Databridge Server | +------------------+--+

1.2 Abbreviations

We use the following abbreviations throughout this guide:

+------------------+--+ | Abbreviation | Name | +==================+==+ | AA | Absolute Address | +------------------+--+ | ABSN | Audit Block Serial Number | +------------------+--+ | AFN | Audit File Number | +------------------+--+ | DASDL | Data and Structure Definition Language | +------------------+--+ | DMSII | Data Management System II | +------------------+--+ | IDX | Index | +------------------+--+ | MCP | Master Control Program | +------------------+--+ | RSN | Record Serial Number | +------------------+--+ | SEG | Segment | +------------------+--+ | WFL | Work Flow Language | +------------------+--+

1.3 Related Documentation

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

+----------------------+---+ | **Databridge product** | **On the Databridge installation image, the** | | **documentation** | DOCS folder contains guides for installation, | | | error codes, and administrator\'s guides for | | | each Databridge product. These documents | | | require Adobe Reader for viewing, which you | | | can download from the Adobe | | | website. This | | | documentation, and current technical notes, | | | is also available on the Attachmate support | | | site. | | | | | | You can also find documentation for | | | Databridge Enterprise Server and the | | | Databridge Client Console in the Help menu. | | | You will need a modern browser to view | | | DBEnterprise documentation. | +----------------------+---+ | **CANDE, KEYEDII,** | **If you are not completely familiar with DMSII** | | **and Unisys MCP** | **configuration, refer to your CANDE, KEYEDII** | | **server** | **and UNISYS MCP documentation.** | | **documentation** | | +----------------------+---+

2. Introducing Flat File Client

Micro Focus Databridge is a combination of host and (optional) client software that provides automated replication of DMSII databases and MCP flat files. All replications occur while the DMSII database is active. After the initial clone, Databridge updates the secondary database, copying only the DMSII data changes from the audit trail.

The Flat File Client (also known as PCSPAN) is simply a Windows implementation of the DBSPAN accessory on the MCP. As is the case with DBSPAN, rather than update the secondary database, the Flat File Client creates data files that contain the data records for the updates. This approach is useful when a Databridge Client does not exist for a particular database or platform or when the data has to be transformed before being loaded into a secondary database.The Flat Client has a very similar architecture to relational database clients such as the SQL Server and the Oracle Clients.

See Chapter 1 in the Databridge Client Administrator\'s Guide, " Introducing Databridge Client", for a description of the Client/Server relationship and the system architecture.

The Flat File Client uses an ODBC database (e.g. SQL Server or mySQL) to store its control tables, which are used to keep track of the state of the replication process and to hold the descriptions of the DMSII and flat file record layouts.

The Flat File Client uses a configuration file that is very similar to the one used by the relational database clients. The main differences are the lack of a bulk loader section and a new section named `PCSpan` used to hold parameters specific to the Flat File Client.

See Client Configuration Files in the Appendix for a description of the Flat File Client configuration file, and Appendix C: Client Configuration in the Databridge Client Administrator's Guide for common parameters with relational database clients.The Flat File Client implements most of the commands that are used by the relational database clients, except for the `generate`, `refresh`, `reorg` and `rowcounts` commands..

3. Getting Started

The material in this chapter supplements Chapter 2 of the Databridge Client Administrator's Guide. Most of that material applies equally to the Flat File Client. This chapter simply points out the differences between two clients. As mentioned previously, the Flat File Client uses Client control tables in exactly the same way as relational database clients do. These control tables are common to all clients and they have exactly the same layout. See Databridge Client Control Tables in the Databridge Client Administrator's Guide for a complete description of these tables.

Before you clone a DMSII data source, you must create the Client control tables. Once they're created, you can populate them with DMSII information sent by the Databridge Engine and create the data files that will store your cloned data. The resulting data files are based on information from the Client control tables and any additional customizations you make.

To customize how DMSII data sets get mapped to their corresponding relational database tables, you can use the Client Configurator or you can write user scripts in SQL. The Client Configurator is compatible with command-line operations and can help make reorganizations easier to handle. For more information on the Client Configurator, go to the Help menu in the Client Console. Writing and testing user scripts is typically time-consuming and requires a bit of trial and error. For instructions, see Customizing with User Scripts in the Databridge Client Administrator's Guide.

In this chapter

3.1 Creating Client Control Tables

The material in the corresponding section in Chapter 2 (Creating Client Control Tables) of the Databridge Client Administrator's Guide applies to the Flat File Client, except that the command line client has no `-W` option and the configuration file parameter `use_nt_authen` is not supported. If you are using a Microsoft SQL database, you can use integrated Windows Authentication when creating the ODBC data source. Because it has no bulk loader, the Client does not need this option.The Flat File Client has a `dbfixup` program that updates your control tables to the current version. It works exactly the way the relational database clients do.

3.2 Defining a Data Source

Refer to the material in Chapter 2 of the Databridge Client Administrator's Guide, Defining a Data Source. All information applies to the Flat File Client except for the data table creation and index creation user scripts (these don't apply because no relational database tables are involved).The Databridge Client Administrator's Guide section on Decoding DMSII Dates, Times, and Date/Times applies to the Flat File Client except for this: the dates are written out using the format specified by the `span_date_format` parameter in the [PCSpan] section of the client configuration file.The Databridge Client Administrator's Guide section on Creating Indexes for Tables does *not* apply to the Flat File Client.The Databridge Client Administrator's Guide section on Adding a Non DMSII Column applies to the Flat File Client; in fact it supports more types of external columns than the relational database clients do in order to remain compatible with DBSPAN. The table below shows the values for all the external column types supported by the Flat File Client.

Note

The value for the Bit column in this table is equal to the value in the dms_subtype column of the DATAITEMS Client control table. The exception is bit 14, which results in a dms_subtype of 0. Bits are numbered from right to left; the right-most bit is 1.

Bit	Value	User Column Name	Description
1	1	update_type	Database update type, as follows: • 0 for extract • 1 for create • 2 for delete (bit 10 must also be enabled) • 3 for modify **NOTE:** You cannot use this value at the same time as bit 11.
2	2	update_time	Time the update was written to the flat file.
3	4	update_ts	N/A
4	8	audit_ts	DMSII audit file timestamp. This column is set to NULL during the initial clone. **NOTE:** You cannot use this bit at teh same time as bit 13.
5	16	audit_filenum	Audit file number
6	32	audit_block	Audit block serial number (ABSN)
7	64	source_name	Data source name
8	128	source_id	Data source identifier as defined in the DATASOURCES Client control table
9	256	my_id	N/A
10	512	deleted_record	N/A
11	1024	update_type	Expanded database update type as follows: 0 for extract 1 for create 2 for delete 3 for modify **NOTE:** • If you reuse the key for this record, inserting the new duplicate record removes the key. • You cannot use this value at the same time as bit 1 or bit 10. Bits 10 and 11 are compared in Preserving Deleted Records. • This bit and bit 1 work in the same way, except that this bit preserves the deleted image.
12	2048	source_id	Data source identifier as defined in the DATASOURCES Client control table (key item)
13	4096	audit_ts	Expanded audit file time. This column contains the DMSII audit file timestamp during updates and the starting time of the data extraction during extraction. **NOTE:**You cannot use this bit at the same time as bit 4.
14	8192	user_column1	Generic user column whose entry is left as NULL.

15	16384	sequence_no	N/A
16	32768	delete_seqno	N/A
17	65536	create_time	Time when the record was created in the flat file (PC time).
18	131072	user_column2	Generic user column whose entry is left as NULL.
19	262144	user_column3	Generic user column whose entry is left as NULL.
20	524288	user_column4	Generic user column whose entry is left as NULL.
21	1048576	change_code	The one letter change code for an update A=Add, D=Delete, M=Modify
22	2097152	strnum	The structure number for the data set.
23	4194304	strname	The data set name.
24	8388608	rectype	The record type for variable format data sets.
25	16777216	modflag	A flag that indicates if an update is part of a MODIFY BI/AI pair. A value of 1 for an ADD or a DELETE, indicates that the ADD was originally the AI or DELETE was originally the BI of a MODIFY BI/AI pair.
26	33554432	mode	The mode of the data set as stored in the ds_mode column of the DATASETS Client control table. 0=CLONE, 1=FIXUP, 2=UPDATE.
27	67108864	fmt_level	The format level for the data set.
28	134217728	audit_seg	The segment number of the audit location within the audit file (audit_seg column in DATASETS Client control table)
29	268435456	audit_inx	The index of the audit location within the segment of the audit file (audit_inx column in DATASETS Client control table)
30	536870912	stacknbr	The stack number of the corresponding task on the MCP.

Note

Values for All the Externals Column Types Supported by the Flat File Client

3.3 The Process Command

The `process` command is the main command of the Databridge Flat File Client. It creates the flat files for all data set tables whose active column is 1 in the corresponding entries of the DATASETS Client control table. Since the `define` command initializes the ds_mode column, all the selected data sets are cloned the first time you run a `process` command.

Note

If you do not select specific data sets in the data set global mapping script, the Databridge Client automatically clones all data sets except for remaps, the restart data set, and the global data set. This operation may take a very long time and require a lot of disk space.

You can schedule the `process` command to update the Databridge flat files. The schedule becomes effective after you run the `process` command for the first time. For more information, see Scheduling Updates in the Databridge Client Administrator's Guide.

To create the Databridge extract, fixup, and update files using the command line Flat File Client's `process` command, you must first make sure that the current directory is set to the working directory you created for this data source. If you start the run from the Client Console, you do not have to deal with such details. You simply initiate a `process` command using the Data Source menu.The remainder of the material in The Process Command section of the Databridge Client Administrator\'s Guide applies to the Flat File Client. You can control the locations of the flat files using the `[PCSpan]` section parameters `extract_dir`, `fixups_dir` and `update_dir`. By default, all files are written to the working directory. If you want to keep them separate, you can specify different directories such as \"extracts\", \"fixups\" and \"updates\". The names normally will be relative to the working directory. Thus in the above example, the client would create three subdirectories in the working directory. The files will have similar names, which can be made to include some metadata by using the file name mask. See the `filename_mask` parameter described in Appendix A.

3.4 The Clone Command

Refer to The Clone Command in Chapter 3: Cloning a DMSII Database of the Databridge Client Administrator's Guide for more information.

4. Tracking Updates to the DMSII Database

This chapter covers the tracking of updates to the DMSII database by the Databridge Flat File Client. The Flat File Client tracks updates to a data set (or a table mapped from a data set) by writing them to a flat file that contains some meta-data columns that identify things like the update type. This is very similar to the concept of history tables that the relational database clients support.In addition, it explains how to deal with DMSII reorganizations.

In this chapter

4.1 Creating the Update Flat Files

The system tracks DMSII updates by adding a record that contains the DMSII record data to the corresponding flat file with some meta-data columns added. Updates that involve the addition, deletion, or modification of a record are represented by a column that specifies the update type and the actual record data. The Flat File Client offers two choices for this column, mainly the numeric update_type column or the single character change_code column that the DBSPAN accessory uses. See the table in Defining a Data Source of this manual for details on these columns.

Extracts, fixups and updates can be written to the same set of files (one per table) or to a different set of files (one set for extracts, one for fixups, and one for updates). The directories where these files reside are defined by the `extracts_dir`, `fixups_dir` and `updates_dir` parameters in the `[PCSpan]` section of the Client Configuration Files. They are described in Client Configuration Files in the Appendix of this manual. If using a single set of files, you will likely want to include the mode column in the files.

You can save updates in the Databridge Flat File Client data files by executing the `process` command each time you want to process the updates, or you can schedule a `process` command. See Scheduling Updates in Chapter 4 of the Client Administrator's Guide for details on scheduling.

4.2 Anomalies That Can Occur In Updates

When the Databridge Flat File Client updates the data files, the following anomalies can occur:

Last quiet point (QPT) in an audit file

When processing an update transaction group since the last quiet point, DBEngine or DBEnterprise does the following when it reaches the end of the last available audit file:

- Aborts the current transaction group so that the updates are rolled back. These are not duplicate updates, but updates that could not be committed. These updates will be reapplied the next time you run a `process` command.
- Sends the Databridge Client a status indicating that the transaction group was rolled back. Upon receiving this status, the Databridge Client does not display any messages. In this case the updates in the transaction group are not appended to the data files.

Host application rolls back changes.

If a host application encounters an error condition while updating the DMSII database, it rolls back all of the changes it made. In this case, DBEngine aborts the updates when it finds the aborted transaction indication in the audit file. DBEngine handles the situation in one of two ways based on the setting of the Engine control file parameter CONVERT REVERSALS TO UPDATES:

1. If CONVERT REVERSALS TO UPDATES is FALSE (the default setting), DBEngine sends an abort transaction status to the Databridge Client and then reprocesses the transaction group, excluding any updates by the program(s) that rolled back its updates. In this case, no data in the aborted transaction is written to the data files.
2. If CONVERT REVERSALS TO UPDATES is TRUE, DBEngine will continue to process the audit file, converting the items marked as reversals to normal updates, in a manner similar to the method employed by DMSII. (DMSII aborts transactions by reversing the updates previously done. Thus an ADD will be reversed to a DELETE, a DELETE reversed to ADD, MODIFY to a MODIFY using the Before Image). All updates, including those that were aborted and their reversals, are written to the data files.

In order to handle these situations, the Flat File Client uses temporary files to hold the updates in a transaction group. Upon getting a COMMIT from the Databridge Engine (we use this term to refer to both DBServer on the MCP and Databridge Enterprise Server), it appends the updates for all the tables that were updated to the corresponding data files. In the case of getting an ABORT from the Databridge Engine, it does not update the data files. In both cases, the temporary files are invalidated so that they get overwritten when the Client gets the next group of updates.

5. DMSII Reorganization and Rollbacks

This section lists changes that can occur to the DMSII database, how those changes affect the Databridge Flat File Client control and data files, and how to handle them in the Client. For instructions on handling a DMSII reorganization on the host, see " Prepare for a DMSII Reorganization" in Chapter 11 of the Databridge Host Administrator\'s Guide.

In this chapter

- Initializations
- Reorganizations
- Rollbacks
- Recloning

5.1 Initializations

A DMSII initialization occurs when a DMSII data set is purged of its records. When a data set is initialized, the Databridge Engine sends the Databridge Client a stateinfo record with a mode value of 4. The Databridge Client performs the actions described below after displaying the following message:

```
DataSet
      name[/
      rectype] has been purged
```

The Client deletes all of the files belonging to this data set and re-creates them, effectively purging the files of all records. When DBEngine is done sending stateinfo records, it sends a status of DBM_PURGE(21), causing the client to display the following message:

```
DataSets purged by DBEngine
```

The normal update processing will repopulate them.

::: {.note} An initialization does not change the data set format level in the DMSII database. :::

5.2 Reorganizations

Although there are basically three types of DMSII database reorganizations (see Chapter 11: DMSII Reorganizations and Rollbacks in the Databridge Host Administrator\'s Guide) --- record format conversion, file format conversions, and garbage collection reorganizations --- the types of reorganizations are not as important as whether the reorganization changes record layouts or record locations, as follows:

- DMSII record format conversions change record layouts. When a data set is affected by a record format conversion, you must apply parallel changes to the flat file layouts. See Managing DMSII Changes to Record Layout in the Databridge Client Administrator's Guide.
- DMSII file format conversions and garbage collection reorganizations change record locations. Only certain data sets require recloning in this case. See DMSII Changes to Record Locations in the Databridge Client Administrator's Guide.

> **Note**
>
> Filler substitutions are handled the same way record format reorganizations are. In a filler substitution, the item count changes.

5.2.1 Managing DMSII Changes to Record Layout {#b1iqosl1}

Use this procedure if a DMSII reorganization changes the layout of records. DMSII record layouts are changed in the following circumstances:

- Record format conversion (also called structural reorganization in this section)
- Filler substitutions

When the Databridge Engine notifies the Databridge Client that the layout of records have changed, the Databridge Client returns a message for each reorganized data set and then prompts you to run a `redefine` command. It then returns an `exit_status` value of 2 (DMSII reorganization). The `redefine` command can determine whether the layout for the data tables have been affected by the DMSII layout change and if the affected data sets need to be recloned. (For more information about what the `redefine` command does, refer to the Databridge Client Administrator's Guide's section About the redefine Command.

5.2.2 To run the redefine command

If you use the Client Configurator to customize the table layouts, skip steps 1 through 4 and run the Client Configurator instead. It will perform the same actions as the `redefine` command, but will also allow you to make customizations.

If the DMSII changes are extensive or complex, we recommend that you back up the Client control tables first, before proceeding.

Modify user scripts as required by the host layout changes.

Run the `redefine` command as follows:

```
dbutility redefine
         datasource
```

If the `redefine` command results in errors because the user scripts were improperly updated, run a `reload` command using the unload file created in step 1 (or the unload file created by the `redefine` command itself, if you skipped step 1) to restore the control tables. Correct the user scripts and rerun the `redefine` command until no errors result.

Unless you use the `filename_mask` parameter to make the file names unique, you will need to move the data files affected by the reorganization to a backup directory, as the new files will have different record layouts. When you are ready, switch the ds_mode column of any data set whose `ds_mode` is 31 back to 2. Unlike the relational database client there is no `reorg` command that does this.

```
dbutility process
      datasource
```

The same principles apply to the user data files which include AA Values used as keys by the applications that process the files created by the Flat File Client. If the ds_mode column in the corresponding row in the DATASETS Client control table is 0, the data set will be recloned, which means that all data files associated with the data set will be repopulated with extracts. Refer to " Recloning" in Chapter 5 of the Databridge Client Administrator\'s Guide for details on when you need to reclone data sets.

5.3 Rollbacks

A DMSII \"rollback\" describes restoring the DMSII database to an earlier point in time to correct a problem. While restoring the DMSII database, replication stops. You must then restart the Client to resume replication. If the Client has processed updates after the DMSII restore point, this replicated data will be wrong. Upon finding bad information in the stateinfo, the Databridge Engine typically returns a message stating that a rollback has occurred. To resolve this problem, the user data files must also be restored (to the DMSII restore point or earlier). If the Client hasn't processed updates after the DMSII restore point, no action is required. This can often be the case as the Client tends to lag behind the DMSII database by several audit files.The easiest way to achieve this is to periodically save off the unprocessed data files and keep track of the stateinfo in case you have to restore the files to their earlier state. You can save the stateinfo by unloading the Client control tables.

5.4 Recloning

Reasons for recloning include the following:

- DMSII reorganization
- DMSII rollback
- An update is not possible (for example, because a table does not have a unique key)

You can use either the `process` or `clone` command to reclone data sets. The `clone` command lets you specify individual data sets on the command. The `process` command automatically reclones all data sets whose active column is 1 and whose ds_mode column is 0. Both commands perform fixups, tracking, and processing updates as needed (unless the `defer_fixup_phase` or the `stop_after_fixups` parameter is set to True). Refer to Recloning in the Databridge Client Administrator's Guide for more details on this topic.

5.4.1 Recloning Individual Data Sets

Use one of the following procedures to reclone data sets.

Set the current directory to the one you created for the data source.

Set the ds_mode column (in the DATASETS Client control table) to 0 for the data sets you want to clone by running a SQL command. If you are recloning all data sets, using the \" `-Y reclone_all` \" options eliminates the need to do this, as the client will update the DATASETS table automatically when this option is used.

Run the `process` command with the -y option, as follows:

```
dbutility process -y
         datasource
```

The `-y` option forces any data sets whose ds_mode is set to 11 or 12 to be recloned, in addition to the recloning data sets whose ds_mode is set to 0. After the data extraction process is complete for the data sets being recloned, flat files whose active columns are set to 1 in their corresponding Client control table (and whose ds_mode is set to 1 or 2) are updated.

Set the current directory to the one you created for the data source.

Set the parameter `defer_fixup_phase` to True to suspend audit file processing. If you don\'t do this, audit files will be processed twice, once for the data set you clone and once for all of the other data sets.

Synchronize the tables by running a `process` command. Synchronization occurs when all data sets reach the same point in the audit trail.

Refer to " Recloning Individual Data Sets" in the Databridge Client Administrator\'s Guide for more details on this topic.

5.4.2 Adding a Data Set {#b1iu7ssa}

Use this procedure to add a data set after you clone the DMSII database. You don't need to reclone the entire database.

Run a relational database query tool and list the contents of the DATASETS Client control table with the following SQL command:

```
select
      dataset_name, active,
      data_source from DATASETS
```

Set the active column for the data set you want to add to the Databridge data tables to 1 (on), as follows:

```
update DATASETS set active=1 where dataset_name='
      datasetname'
```

Run a `redefine` command.

Run one of the following commands to populate the new tables that correspond to the new data set

```
dbutility process
      datasource
```

--or--

```
dbutility clone
      datasource
      datasetname
```

Note

If you run a `process` command, the Databridge data tables whose active columns are set to 1 in their corresponding Client control table are also updated.

After you complete this procedure, update your data set global mapping script (`script.user_datasets.      datasource`) so that you do not lose this change the next time you run a `define` command.

6. Resources

6.1 Appendix

This appendix lists the configuration parameters for Databridge Flat File Client. Refer to Appendix C: Client Configuration in the Databridge Client Administrator\'s Guide for a complete description of Client Configuration files.

In this chapter

6.1.1 Client Configuration Files

The Databridge Flat File Client uses binary configuration files. Binary configuration files are compatible with both service-initiated operations and command-line operations. However, if you use the service, you must use binary configuration files. (Command-line operations can use either binary or text configuration files, although we recommend always using binary configuration files. The configuration file is named \" `dbridge.cfg` \" and it resides in the \" `config` \" subdirectory of the data source\'s working directory.

6.1.2 Command-Line Options

The following command-line options have no equivalent configuration parameter:

+-----------------+-------------------+------------------------------+ | Option | dbutility Command | Description | +=================+===================+==============================+ | `?` | | Short help | +-----------------+-------------------+------------------------------+ | `-d` | All | Default tracing | +-----------------+-------------------+------------------------------+ |

```
-f              | All                | Specifies the configuration   |
|               |                    | filename                         |
|        filename |                  |                                  |
```

| | | | +-----------------+-------------------+------------------------------+ | `-h` | | Long help | +-----------------+-------------------+------------------------------+ | `-k` | `reload` | Makes the command preserve | | | | the stateinfo of data sets | | | | that have a ds_mode of 2 and | | | | have not been reorganized | +-----------------+-------------------+------------------------------+ | `-m` | All | Includes a 5-digit | | | | millisecond timer in all | | | | output messages | +-----------------+-------------------+------------------------------+ | `-r` | `redefine` | Toggles the setting of the | | | | parameter `use_dbconfig`, | | | | which determines whether the | | | | command uses user scripts | +-----------------+-------------------+------------------------------+ | `-t mask` | All | Tracing options | +-----------------+-------------------+------------------------------+ | `-u` | `configure`, | Unconditionally performs the | | | `define`, | requested command, overriding | | | `redefine` | any warnings that would be | | | **and** `dropall` | displayed without this | | | | option. | +-----------------+-------------------+------------------------------+ | `-v` | All | Causes the client to log | | | | and/or print some additional | | | | messages. | +-----------------+-------------------+------------------------------+ | `-w` | `clone` **or** | Toggles the setting of the | | | `process` | `use_dbwait` parameter. | +-----------------+-------------------+------------------------------+ | `-x` | `clone` | Clones all active data sets | | | | except those specified at the | | | | command line. | +-----------------+-------------------+------------------------------+ | `-y` | `process` | Instructs the client to | | | | reclone all data sets whose | | | | ds_mode has a value of 11 or | | | | 12. | +-----------------+-------------------+------------------------------+ | `-z` | `clone` **or** | Instructs dbutility to not | | | `process` | update the relational | | | | database during a `clone` or | | | | `process` command. | | | | | | | | This option is useful in | | | | determining how much | | | | non-database time and file | | | | I/O is required to extract | | | | data for a data set. | +-----------------+-------------------+------------------------------+ | -B | `display` | Causes the `display` command | | | | to quit after displaying the | | | | DATASETS client control table | | | | records. | +-----------------+-------------------+------------------------------+ | `-F afn` | `process` | Use this option make the | | | | client act as if a | | | | `QUIT AFTER afn` command had | | | | been executed. It applies to | | | | `process` and `clone` | | | | commands only. The range of | | | | values allowed is 1-9999. | +-----------------+-------------------+------------------------------+ | `-K` | `process` | Prevents the audit file | | | | removal WFL from being run on | | | | the mainframe after the | | | | Engine finishes processing an | | | | audit file. | +-----------------+-------------------+------------------------------+ | `-L` |

All | Forces the Client to start | | | | using a new log file. | +-----------------+-------------------+-------------------------------+ | `-O                |`
`All                  | Specifies the ODBC data        |`
`|                    |                        | source to connect to.          |`
`|  ODBCdatasource |                        |                                |`
`|` | | | +-----------------+-------------------+-------------------------------+ | `-P                 | All                  | Sets the password associated  |`
`|                    |                        | with the user ID for the       |`
`|          password |                        | relational database. The       |`
`|` | | password is limited to 30 | | | | characters. | +-----------------+-------------------+-------------------------------+ | `-R` | `redefine` | Forces all data sets to be | | | | redefined. | +-----------------+-------------------+-------------------------------+ | `-T` | All | Forces the program to create | | | | a new trace file when tracing | | | | is enabled. | +-----------------+-------------------+-------------------------------+ | `-U                 | All                      |`
`Specifies the user ID for the |`
`|           userid |                        | relational database. The user |`
`|` | | ID must have the appropriate | | | | resource privileges for the | | | | designated relational | | | | database. | +-----------------+-------------------+-------------------------------+ | `-X` | `define, redefine | Specifies the host password.  |`
`|                    | , clone, process` | | +-----------------+-------------------+-------------------------------+ | `|`process`| Causes all active data sets    |`
`| -Y reclone_all` | | to be recloned. | +-----------------+-------------------+-------------------------------+

6.1.3 Syntax

Follow these conventions in the configuration file:

- For hexadecimal values, use the 0xnnnn format.
- A semicolon (;), except within double-quoted strings, indicates that the remainder of the current line is a comment.
- Enclose section headers in square brackets.
- Section headers and parameter names are not case-sensitive.
- Spaces and tabs between entries are ignored; however, spaces within double quoted values (for example, password values) are read.
- If you are not using a parameter, either comment the parameter out or delete the corresponding line in the configuration file. Do not leave an uncommented parameter without a value after the equal sign (=). Doing so results in syntax error.

You can specify some of these parameters only in the Client configuration file. Other parameters have equivalent command-line options and environment variables. For a complete list of configuration file parameters, their equivalent command-line options, and their related Client command, see [params].

6.1.4 Sample Flat File Client Configuration File

You can view the configuration file by using the `export` command. Refer to the Databridge Client Administrator\'s Guide for details on the `export` command.

To use a parameter that is commented out, delete the semi-colon (;) and after the equals sign (=) enter a value that is appropriate for your site. Boolean parameters can be represented by True or False.

In the example below, some of the commented-out parameters have a value of -1. These parameters include the DBEngine control file parameters that can be overridden by the client (commit frequency parameters and engine workers). This value indicates that the corresponding parameter in the Databridge Engine (or Server) control file will not be overridden by the Client. Do not uncomment these lines, unless you want to supply an actual value. Otherwise, the Client will issue an error.

```
;
; Databridge Client version 6.6 Flat File configuration file -- generated programmatically
;

[Signon]
;user                    = USERID
;password                = PASSWORD
;datasource              = DATASOURCE
;hostpasswd              = HOSTPASSWD

[Log_File]
file_name_prefix         = "db"
;max_file_size           = 0
logsw_on_size            = false
logsw_on_newday          = false
newfile_on_newday        = true
```

```
single_line_log_msgs    = false

[Trace_File]
file_name_prefix        = "trace"
;max_file_size          = 0

[PCSPan]
;
;  (1) DMSII data redaction parameters
;
redacted_database        = 0
;
;  (2) Flat File client parameters
;
add_item_names          = false
add_nl_chars            = false
decimal_char            = "."
delimiter               = ","
extracts_dir            = ""
filename_mask           = csv
fixups_dir              = ""
format                  = csv
ltrim_zeroes            = true
prefix_columns          = 0x00000000
quote                   = "'"
real_format             = scientific,11,6
rtrim_spaces            = true
span_date_delim         = ""
span_date_format        = 21
span_date_scale         = 0
treat_real_as           = real
updates_dir             = ""
use_plus_sign           = false
use_quotes              = true
;
[Params]
;
;  (1) define/redefine command parameters
;
allow_nulls             = true
default_user_columns    = 0x00000000
enable_dms_links        = false
;external_column[n]     = ["name"][,[sql_type][,[sql_length][,"default"]]]
extract_embedded        = false
flatten_all_occurs      = false
force_aa_value_only     = 0
miser_database          = true
read_null_records       = true
sec_tab_column_mask     = 0x00000000
split_varfmt_dataset    = false
strip_ds_prefixes       = false
suppress_new_columns    = false
suppress_new_datasets   = true
use_binary_aa           = false
use_column_prefixes     = false
use_date                = true
use_decimal_aa          = false
use_nullable_dates      = true
;
; (2) process/clone command parameters
;
alpha_error_cutoff      = 10
automate_virtuals       = false
century_break           = 50
;commit_absn_inc        = -1
;commit_longtrans       = -1
;commit_time_inc        = -1
;commit_txn_inc         = -1
;commit_update_inc      = -1
controlled_execution    = false
convert_ctrl_char       = false
;convert_reversals       = -1
correct_bad_days        = 0
dbe_dflt_origin         = direct
defer_fixup_phase       = false
discard_data_errors     = false
display_bad_data        = false
enable_doc_records      = false
;engine_workers         = -1
error_display_limits    = 10,100
inhibit_8_bit_data      = false
inhibit_console         = false
inhibit_ctrl_chars      = false
keep_undigits           = false
linc_century_base       = 1957
max_discards            = 0,100
max_retry_secs          = 20
max_srv_idle_time       = 0
max_wait_secs           = 3600,60
min_check_time          = 600
n_dmsii_buffers         = 0
n_update_threads        = 8
null_digit_value        = 9
numeric_date_format     = 23
preserve_deletes        = false
set_blanks_to_null      = false
set_lincday0_to_null    = false
```

```
show_perf_stats         = true
show_statistics         = true
show_table_stats        = true
sql_exec_timeout        = 180,0
statistics_increment    = 100000,10000
stop_after_fixups       = false
stop_after_gc_reorg     = false
stop_after_given_afn    = false
stop_on_dbe_mode_chg    = false
track_vfds_nolinks      = true
use_dbwait              = false
use_latest_si           = false
;
; (3) Server options
;
;shutdown {until | for} hh:mm after stop
;stop {before | after} task "name"
;stop {before | after} time hh:mm[:ss]
;
; (4) miscellaneous command parameters
;
display_active_only     = true
;
; (5) user scripts
;
user_script_bu_dir      = ""
user_script_dir         = "scripts"
;
; (6) external data translation parameters
;
use_ext_translation     = false
eatran_dll_name         = "DBEATRAN.DLL"

[Scheduling]
;
; dbutility process command only
;
;daily                  = 08:00, 12:00, 17:00, 24:00
;exit_on_error          = false
;sched_delay_secs       = 600
;sched_minwait_secs     = 3600
;sched_retry_secs       = 3600

[EbcdicToAscii]
; e1 = a1
; e2 = a2
;  ...
; en = an
;

[DBConfig]
default_date_fmt        = 21
```

6.1.5 Processing Order

Configuration file options override environment variables. Command-line options override both environment variables and configuration file options.

The parameter processing order is as follows:

1. The operating system login name (user ID) is used as the lowest level default for the database user ID
2. Environment variables (DBUSERID, DBPASSWD, and DBHOSTPW).
3. Command-line options `-d` (for full tracing), `-v` (for verbose messages), `-t` (for creating a Databridge Client trace file) and `-T` (for forcing the client to start a new trace file), and `-f` (for specifying a configuration file other than the default `dbdridge.cfg`). These options are processed in the order in which they appear on the command line.
4. Parameters specified in the configuration file. You can specify the configuration file via the `-f` option. If you do not specify a configuration file name via the `-f` option, the Databridge Client tries to open the default configuration file (`dbridge.cfg` in the config subdirectory of the data source\'s working directory); if the file does not exist, the Databridge Client uses the default values for each configuration file parameter. The absence of a configuration file is not treated as an error only when running the command-line client. If you use the service or daemon, the absence of a configuration file named `dbridge.cfg` is treated as an error.
5. All remaining command-line options. In the final pass, a command-line option with a configuration file equivalent overrides the configuration file entry.

6.2 Parameter Descriptions

6.2.1 [signon]

Use the `[signon]` section of the dbridge.cfg file to enter information for signing on to the relational database and Databridge Server on the host.

The configuration file must include the ODBC data source, signon parameters to access the relational database, and a user and a password (unless you use a SQL Server database with Integrated Windows authentication, in which case the userid and password can be omitted).

When using the Client Console, these parameters are supplied at the time the data source is created. You need to right-click on the service in the tree view and select **Add Data Source** from the pop-up menu to get the dialog that allows you to supply these parameters.

Parameter	Description
`datasource`	Default: None Command-line option: `-O` This parameter is the name that identifies the ODBC data source used to access the ODBC database that holds the Client Control tables. This ODBC data source is configured using the Control Panel during the Client installation procedure.

`hostpasswd`	Default: None
	Range: 17 alphanumeric characters
	Command-line option: `-X`
	
	Use the host password parameter to specify the password associated with Databridge Server on the host. This parameter must match exactly the KEY parameter defined in the Host Server control file. For example:
	
	`DBServer`
	`&nbspKEY = "Secret"`
	`dbridge.cfg`
	`&nbsphostpasswd = Secret`

`password`	Default: None Command-line option: `-p` Use the password parameter to specify the password associated with the user ID for the relational database. The password must be valid for the user ID or the connection to the relational database server will fail. Passwords are limited to 30 characters. If your password contains non alphanumeric characters other than the underscore, you must enclose the password in double quotes, as follows: `password = "a$bb%"` **NOTE:**The password is always encoded in both text and binary versions of the client configuration file. For more information, see “Export or Import a Configuration File” in the Databridge Client Administrator's Guide or see “Export the Client Configuration to a File” in the Databridge Client Console Help. Passwords communicated between the Databridge Client and Databridge Server are not encoded.
`user`	Default: Login userid Command-line option: `-U` Use the user parameter to specify the user ID for the relational database. The user ID must have the appropriate resource privileges for the designated relational database, as explained in [Setting Up a User ID](NEED LINK) in the Databridge Installation Guide.

[signon] parameters with equivalent environment variables

[signon] Parameter	Environment Variable	Option	dbutility Command
`datasource`		`-O`	All
`hostpasswd`	DBHOSTPW	`-X`	`define` , `redefine` , `process` , `clone` , and `switchaudit`
`password`	DBPASSWD	`-P`	All
`user`	DBUSERID	`-U`	All

6.2.2 [Log_File]

This section is the same as in the regular Databridge Clients. Refer to "" in "Appendix C: Client Configuration" of the Databridge Client Administrator's Guide.

6.2.3 [Trace_File]

This section is the same as in the regular Databridge Clients. Refer to [Trace_File] in Appendix C: Client Configuration of the Databridge Client Administrator's Guide.

6.2.4 [PCSpan]

This section contains parameters specific to the Flat File Client. When using the Client Console or the Client Configurator, these parameters are accessed using the PCSpan or the PCSpan > Advanced Parameters nodes in the Client Configuration tree view. Names that follow these appear in parentheses and are either labels of controls or of groups of controls. In some instances where the names are very different, we provide both the control labels and the groups' names.

redacted_database

Default:0

Range:0-2

Maintain this parameter at 0 unless you would like to configure data redaction. Contact customer care on configuring data redaction.

add_item_names

Default: False

Range: True or False

Configurator: PCSpan (Options > Insert Column Names Record)

The `add_item_names` parameter is a boolean that adds a record with the DMSII item names to the start of all flat files. The names are the names used in the DASDL, except for items that have OCCURS clauses. The latter get the suffix " `[n]` " appended to the names to make the names unique, where *n* is the occurrence number of the item starting at 1. For example an item named XXX with the clause "OCCURS 5 TIMES" will result in columns named XXX[1] through XXX[5].

add_nl_chars

Default: False

Range: True or False

Configurator: PCSpan (Options > Add New Line Characters ...)

The `add_nl_chars` parameter is a boolean that causes the client to automatically add a CR LF to the end of fixed length records. This parameter is ignored the case of variable length records, as they must end with new-line characters.

decimal_char

Default:"."

Range:"." or ","

Configurator: PCSpan (Special Characters > Decimal Point Character)

The `decimal_char` parameter specifies the decimal point character to be used in numeric items. It defaults to "." and the only other allowable value is "," to be used for European sites.

delimiter

Default:","

Range:1 character string

Configurator: PCSpan (Special Characters > Column Delimiter)

The `delimiter` parameter defines the character that is used to separate items. This parameter is only meaningful when using CSV or COMMA formats. It defaults to ",". You can use a variety of characters as the delimiter, including the vertical bar and the TAB character. To specify tab you can use "\t" or simply TAB.

extracts_dir

Default:""

Range: String

Configurator: PCSpan (Directories)

The `extracts_dir` parameters allows you to place the extract records in a separate directory from fixup and update records. The default value of \"\" corresponds to the working directory for the data source. Typically you would set these to "extracts", "fixups" and "updates" to keep them separate. File names that contain special characters, must be enclosed in double quotes and if the name contains a backslash, you must use two backslashes (the single backslash is used as the force character, e.g. "\t" represents a TAB).

filename_mask

Default:0

Range:0..7

Configurator: PCSpan > Advanced Parameters (File Name Mask)

The `filename_mask` parameter allows you to include any combination of the starting AFN, ABSN and TIMESTAMP in the fixups and updates flat file names. Once the filename suffix has been established it remains in effect for the duration of the run. A bit mask of 1 indicates AFN, a mask of 2 indicates ABSN and a mask of 4 indicates TIMESTAMP. A value of 7 specifies all three.

fixups_dir

Default:""

Range: String

Configurator: PCSpan (Directories)

The `fixups_dir` parameter allows you to place the fixup records in a separate directory from extract and update records. The default value of \"\" corresponds to the working directory for the data source. Typically you would set these to "extracts", "fixups" and "updates" to keep them separate. File names that contain special characters, must be enclosed in double quotes and if the name contains a backslash, you have to use two backslashes, as \ is used as the force character (e.g. "\t" represents a TAB).

format

Default:CSV

Range:CSV, FIXED or COMMA

Configurator: PCSpan (Record Format)

The `format` parameter is only used to determine if the records are fixed-length or not, as it is nothing more than a collection of parameter settings. The three formats supported are: CSV format, FIXED format and COMMA format. The last two formats are meant to mimic COMMA FORMAT and FIXED FORMAT in DBSupport. While CSV format closely resembles VARYING FORMAT in DBSupport.

ltrim_zeroes

Default: True

Range: True or False

Configurator: PCSpan (Options)

The `ltrim_zeroes` parameter controls whether or not to strip leading zeroes for numbers in CSV format.

prefix_columns

Default:0

Range: Bit mask

Configurator: PCSpan > Advanced Parameters (Prefix Column Mask)

The `prefix_columns` parameter specifies a mask that is ANDed with the `default_user_columns` mask to determine which external columns are to be added to the start of the record (the remaining columns are added at the end of the record, just like in the standard Clients). The bits defined for the parameter `default_user_columns` were expanded to include the items that DBSupport allows. Some of the columns used by the standard Clients are not meaningful for the Flat File Client and are therefore not supported. Attempting to set them in the `default_user_columns` specification has no effect.

quote

Default:"""

Range:""" or "\""

Configurator: PCSpan (Special Characters > Quote Character)

The `quote` parameter allows you to use single or double quotes to enclose character data. This is only meaningful for COMMA and CSV formats.

real_format

Default: Scientific, 11, 6

Range: Scientific or Decimal

Configurator: PCSpan (Format DMSII REAL as)

The `real_format` parameter has 3 arguments, which specify the format type, the length and the scale to use when formatting a DMSII REAL in the flat files. The format type can be either scientific or decimal, just like in DBSupport. The remaining two arguments are the length and scale, which allow you to further control the format used to write the data to the flat file.

rtrim_spaces

Default: True

Range: True or False

Configurator: PCSpan (Options)

The `rtrim_spaces` parameter controls whether or not to strip trailing blanks in CSV format. COMMA format does not trim spaces, as it treats character data as fixed length. Furthermore, it does not do anything special with quote characters in the data that is enclosed in quotes, while CSV format uses two quote characters to represent a quote in the data (see the `use_quotes` parameter below). This parameter can be augmented by the `[Params]` section parameter `set_blanks_to_null`, which causes blank items to be represented as NULL (i.e. the empty string ",,").

span_date_delim

Default:\"/\"

Range:\"/\", \"-\", \".\" or \"\"

Configurator: PCSpan > Advanced Parameters (Dates)

The `span_date_delim` parameter allows the year, month and day to either be separated by slashes, dashes, or dots (or not separated at all). If you use a date delimiter, then the hours, minutes and seconds will be separated by colons and the time part will be preceded by a blank. If the delimiter is the empty string, then they are all run together to form a 14-digit number.

span_date_format

Default:21

Range:21 .. 26

Configurator: PCSpan > Advanced Parameters (Dates)

The `span_date_format` parameter, in conjunction with the `span_date_delim` and `span_date_scale` parameters, defines how dates are represented in flat files. The format uses the same date format codes as the standard Client. Currently, only the format with 4-digit years is supported.

span_date_scale

Default:0

Range:0 .. 9

Configurator: PCSpan > Advanced Parameters (Dates)

The `span_date_scale` parameter specifies how many digits after the seconds are to be added to represent fractions of seconds. When the scale is greater than zero a decimal character (dot or comma) is added after the seconds.

treat_real_as

Default:real

Range:real, time6 or binary

Configurator: PCSpan > Advanced Parameters (Treat DMSII REAL as)

The `treat_real_as` parameter indicates how the `define` and `redefine` commands are to treat all REAL items that are not customized. This parameter allows you to do this without the need to do any customization using SQL user scripts. There are 3 choices available: (1) a value of `real` indicates that a REAL should be treated as a floating point number; (2) a value `time6` indicates that a REAL should be treated as a TIME(6) value; (3) a value of `binary` indicates that a REAL should be written out to the flat files as 12-hex digits, just like an RSN. If you have REAL items in DMSII that are timestamps, you need to treat them as BINARY (if you know that they are TIME(6) data, you can treat them as such). If you just treat them as REAL, the most significant bit of the item will be lost when the number is converted to scientific or decimal format, as this bit is unused for a floating point values on the A-Series machines. On the other hand timestamps use this bit, which is typically part of the date.

updates_dir

Default:\"\"

Range: String

Configurator: PCSpan (Directories)

The `updates_dir` parameter allows you to place the update records in a separate directory from extract and fixup records. The default value of \"\" corresponds to the working directory for the data source. Typically you would set these to "extracts", "fixups" and "updates" to keep them separate. File names that contain special characters, must be enclosed in double quotes and if the name contains a backslash, you have to use two backslashes, as \ is used as the force character (e.g. "\t" represents a TAB).

use_plus_sign

Default: False

Range: True or False

Configurator: PCSpan (Options > Use + for Positive Signed Numbers)

The `use_plus_sign` parameter is a boolean that controls whether or not positive signed numbers are always preceded by a + sign.

use_quotes

Default: True

Range: True or False

Configurator: PCSpan (Options)

The `use_quotes` parameter is a boolean that specifies whether or not character data should be enclosed in quotes. If this parameter is set to False the `quote` parameter is ignored. This parameter is meaningless when using FIXED format.

6.2.5 [params]

See Appendix C: Client Configuration of the Databridge Client Administrator's Guide for a description of these parameters. Note that several of the relational database Client parameters do not apply to the Flat File Client. We list these parameter below and only provide details if they are different than what is in Appendix C of the Databridge Client Adminstrator's Guide.

Define and Redefine Command Parameters

DEFAULT_USER_COLUMNS

Default:0

Range:0 -- 16383 (Some bit combinations are not allowed)

Configurator: Customizing > User Columns

The `default_user_columns` parameter adds non-DMSII columns (user columns) to all data sets in the client database. The Flat File Client supports more non-DMSII columns than the relational database Clients, see `external_columns` below for a complete list of these columns.

EXTERNAL_COLUMN[N]

Default:N/A

Range:N/A

Configurator: Customizing > User Columns

This parameter allows you to globally change the column name, sql_type, or sql_length of the non-DMSII columns described in *Numeric Date and Time in Non-Contiguous Columns*. The syntax is as follows:

```
external_column[n] = ["name"[,sql_type[,sql_length]]]
```

Where	Is
n	The corresponding bit number (dms_subtype value) for the non-DMSII column. **NOTE:** The brackets and value are required syntax.
name	Custom column name
sql_type	An integer value that represents the internal code for the SQL type that you want to use. The program only accepts data types that make sense for a particular column. For instance, you cannot set the AFN to a bit or a char, but you can set it to an int or a dec(10).
sql_length	A value that represents the length of the data type. Specify this value only if the data type requires it. If the data type does not have a length specification, specifying a value may cause an error.

For example, the entry below causes the audit_filenum column to be renamed AFN (the double quotation marks are optional since no special characters are involved); the sql_type and sql_length remain unchanged.

```
external_column[5]    = "AFN"
default_user_columns  = 0x0010
```

In the example below, the data type of the audit_block column changed to dec(12).

```
external_column[6]    = ,11,12
```

DEFINE AND REDEFINE COMMAND UNCHANGED PARAMETERS

- allow_nulls
- enable_dms_links
- extract_embedded
- flatten_all_occurs
- force_aa_value_only
- miser_database
- read_null_records
- sec_tab_column_mask
- split_varfmt_dataset
- strip_ds_prefixes
- suppress_new_columns
- suppress_new_datasets
- use_binary_aa
- use_date
- use_column_prefixes
- use_decimal_aa

Process and Clone Command Parameters {#b1iumwl7}

PROCESS AND CLONE COMMAND UNCHANGED PARAMETERS {#B1JL0WWD}

See "Process and Clone Command Parameters" in Appendix C: Client Configuration of the Databridge Client Administrator's Guide for details.

- alpha_error_cutoff
- automate_virtuals
- aux_stmts
- century_break
- commit_absn_inc
- commit_longtrans
- commit_time_inc
- commit_txn_inc
- commit_update_inc
- convert_ctrl_char
- convert_reversals
- correct_bad_days
- dbe_dflt_origin
- defer_fixup_phase
- discard_data_errors
- display_bad_data
- enable_doc_records
- engine_workers
- error_display_limits
- inhibit_8_bit_data
- inhibit_console
- inhibit_ctrl_chars
- linc_century_base
- keep_undigits
- max_discards
- max_retry_secs
- max_srv_idle_time
- max_wait_secs
- months
- n_dmsii_buffers
- n_update_threads
- null_digit_value
- numeric_data_format
- set_blanks_to_null
- set_lincday0_to_null
- show_perf_stats
- show_statistics
- show_table_stats
- sql_exec_timeout
- statistics_increment
- stop_after_fixups
- stop_after_gc_reorg
- stop_on_dbe_mode_chg
- track_vfds_no links

- use_dbwait
- use_latest_si

Server Option Parameters

See Appendix C of the Client Administrator's Guide.

Display Command Parameters

See Appendix C of the Client Administrator's Guide.

User Scripts Parameters

See Appendix C of the Client Administrator's Guide.

6.2.6 [Scheduling]

See Appendix C of the Client Administrator's Guide.

6.2.7 [EbcdictoAscii]

See Appendix C of the Client Administrator's Guide.

6.2.8 External Data Translation DLL Support

The Flat File Client does not currently use a data translation DLL, however, we could implement such a DLL to create UTF8 files, if the need arises.

7. Glossary of Terms

absolute address (AA) value

AA is a DMSII term that stands for absolute address. An absolute address value is an A Series WORD (48-bits in length). In the Databridge Client, AA is the hexadecimal representation (12 character strings containing the characters 0--9 and A--F) of the AA value on the host. Databridge Client uses the AA values to implement unique keys for the parent structures of embedded data set records. It also uses AA values to reference the records of data sets that do not have DMSII SETS with the NO DUPLICATES ALLOWED attribute.

AA values are not constant. Any DMSII reorganization (record conversion, file format, or garbage collection) changes these values.

Databridge Client supports numeric AA values that are stored as NUMBER(15) in Oracle and BIGINT in SQL Server. It also supports binary AA values that are stored as RAW(6) in Oracle and BIN/ARY(6) in SQL Server.

Audit Files

An audit file is created by DMSII and contains the raw format of changes made to the DMSII database by update programs. Audit file records contain the deletes, adds, and modifies that were made to the various structures. It can contain, for example, hours\', days\', or weeks\' worth of information.

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.

If you set the DBEngine Read Active Audit option, Databridge can access the current audit file. If you do not set Read Active Audit = true in the DBEngine parameter file, Databridge can access audit information up to and including the current audit file minus one. The audit file contains the update level at the time the audit file was created. The update level in the audit file and the update level in the DESCRIPTION file used by Databridge must match before Databridge will update a replicated database.

When an audit file is closed, DMSII creates the next one in the series. Audit files are closed for several reasons, including the following:

- An operator closes the audit file with the mixnumber SM AUDIT CLOSE command.
- The audit file reaches the file size set in its DASDL.
- There is an I/O error on the audit file.
- There is not enough disk space for this audit file.
- The database update level changes due to database definition changes
- A Databridge accessory closed the file in preparation for the fixup phase after extracting records from a DMSII database.
- The current audit file could not be found.
- A file reorganization was executed to modify the DMSII structure.

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.

Batch Console

The Batch Console automates routine Client tasks by allowing command files/shell scripts launched by the Databridge Client Manager to interact with the service.

caching

A process that filters files before they\'re requested by the Databridge Client. Caching allows Databridge Enterprise Server to send Client data requests quickly and without placing an additional resource burden on the mainframe.

client

The client is the computer system that will receive DMSII records from the primary database. The client could be a Windows computer, a UNIX computer, or an MCP server. The client can have a relational or a DMSII database.

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.

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 Director

Databridge Director (also referred to as DBDirector) is a Windows Service installed with Enterprise Server that starts Enterprise Server whenever a connection request is received.

When you start your computer, DBDirector starts and reads the ListenPort registry value to determine which TCP/IP port communicates with Databridge Clients.

Databridge Server

Databridge Server is a generic term that can refer to either DBServer or DBEnterprise Server. The two are interchangeable as far as the Databridge Client is concerned.

DBClient

dbutility

DBClient (vs dbutility)

DBClient (vs dbutility)

data_sources

DBClient (vs dbutility)

A Client program that is launched by the Client Manager service. DBClient handles the processing of DMSII data and updates the same as dbutility, except that it runs as a background run and uses the Client Console to display its output and interact with the operator.

DBClntCfgServer

A program that handles all requests from the Client Console specific to a data source. These requests include updating the Client configuration file, providing access to the Client control tables, and handling the Client Configurator. Like DBClient, this program is run by the Client Manager service as a background run.

DBClntCfgServer

A program that handles all requests from the Client Console specific to a data source. These requests include updating the Client configuration file, providing access to the Client control tables, and handling the Client Configurator. Like DBClient, this program is run by the Client Manager service as a background run.

DBServer

DBServer is a Databridge Host accessory that responds to Databridge Client requests for DMSII data or DMSII layout information and provides communications between the following components:

- DBEngine and Databridge Enterprise Server
- DBEngine and the Databridge Client

> **Note**
>
> When Enterprise Server is used with the Databridge Client, Enterprise Server takes over much of the functionality of DBServer and DBEngine.

direct disk

A replication method that allows Databridge Enterprise Server to clone and track DMSII data sets without using any significant mainframe resources. Direct disk replication requires a SAN (Storage Area Network) or Logical Disks configured to make MCP disks visible in Windows.

entry point

A procedure in a library object.

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).

file format conversion

A type of DMSII reorganization affects file size values (for example, AREASIZE, BLOCKSIZE, or TABLESIZE), but it does not change the layout of the records in a DMSII database.

flat files

A flat file is a plain text or mixed text and binary file which usually contains one record per line. Within the record, individual fields may be separated by delimiters, such as commas, or have a fixed length and be separated by padding. An example of a flat file is an address list that contains fields for *Name*and *Address*.

garbage collection reorganization

A garbage collection reorganization moves records around, but it doesn't change the layout of the DMSII database. Its primary function is to improve disk and/or I/O efficiency by eliminating the space occupied by deleted records. Optionally, a garbage collection reorganization reorders the remaining records in the same sequence as one of the sets.

lag time

The lag time is defined as the elapsed time between the time a record in the DMSII database is updated and the time where this update appears in the relational database. This value accounts for any difference between the clock on the mainframe and that on the client machine.

mutex

A mutex is an operating system resource that is used to implement a critical section and prevent multiple processes from updating the same variables at the same time.

null record

A record for a data set where every data item is null.

null value

The value defined in the DASDL to be NULL for a data item. If the DASDL does not explicitly specify a NULL value for a data item, the NULL value is all bits turned on.

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.

record format conversion

A type of DMSII reorganization that occurs when a data set or set (group of keys) is reordered or reformatted. It indicates that changes were made to a data set format, or to data items, such as changing the length of an item, for example, BANK-ID NUMBER (10) to BANK-ID NUMBER (15).

record serial number (RSN)

Record sequence numbers (RSN) are 48-bit quantities used by the Databridge Engine, in the case of DMSII XE, to uniquely identify a record. RSNs will always be used instead of AA values when available except for data sets having embedded data sets. RSNs are always static; they will not change after a garbage collection reorganization.

reorganization

Structural or formatting changes to records in the DMSII database, which may require parallel changes to (or re-cloning of) records in the secondary, or relational, database. See also glossentry_title and glossentry_title.

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.

replication

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

rollback

A systematic restoration of the primary or secondary database to a previous state in which the problem or bad data is no longer found.

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.

semaphores

Operating system resources that are mainly used to implement thread synchronization and signaling.

service

The service (Windows) or daemon (UNIX) that automates most Client operations. It handles operator requests from the Client Console and routes all log and informational messages to the consoles.

set

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

state information

Data stored in the DATASETS Client control table that reflects information about each of the cloned data sets. This information includes the current replication phase (MODE) of the data set. When the data set is successfully extracted, the information also includes the location in the DMSII audit trail from which the last group of updates for the data set was read. This consists of the audit file number (AFN), the audit block sequence number (ABSN), the segment number (SEG) and the index (INX) of the block within the audit file and the timestamp of the block in the audit trail.

structure

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

table

A data structure in the client database corresponding to a data set or remap in the host DMSII database.

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.

8. Legal Notice

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

Contains Confidential Information. Except as specifically indicated otherwise, a valid license is required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

For information about legal notices, trademarks, disclaimers, warranties, export and other use restrictions, U.S. Government rights, patent policy, and FIPS compliance, see https://www.microfocus.com/about/legal/.