# MICRO FOCUS®

# Client for Kafka Administrator's Guide

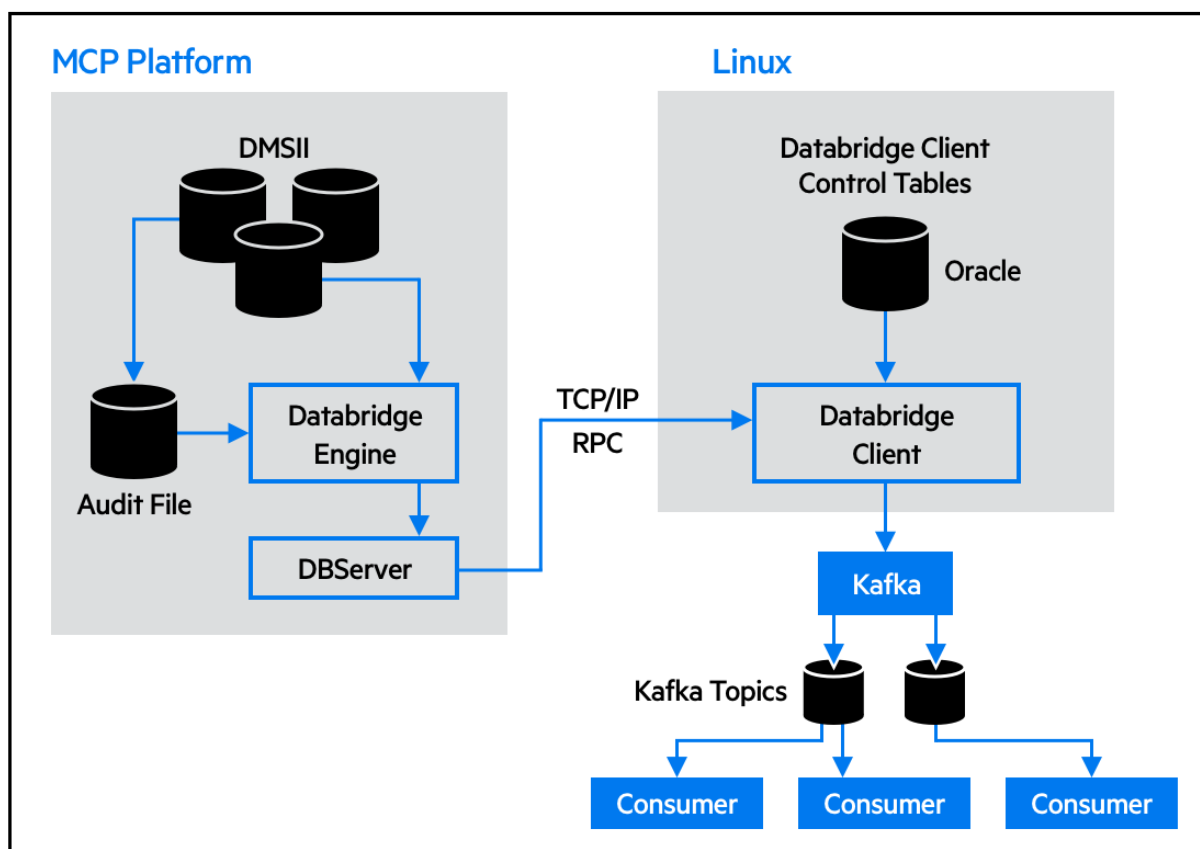**7.0**

# Table of contents

# 1. Introduction

## 1.1 Introducing the Databridge Client for Kafka

The Databridge Client for Kafka enables the ability to utilize the Kafka messaging system within the Databridge architecture. The Kafka messaging system is a scalable fault-tolerate data management system that provides efficient real-time data processing.

The Databridge Client for Kafka acts as a Kafka Producer. **Producers** in the Kafka environment publish **messages** which can be configured to be included in **topics** dispersed by **brokers**.

Communication to Kafka can optionally be authenticated using Kerberos and can use SSL/TLS to encrypt the Kafka datastream if desired.

# Databridge Client for Kafka



## 1.2 Kafka Overview and Roles

The Databridge Client for Kafka utilizes an open source tool that provides real-time messaging and data processing. The Databridge Client for Kafka acts as a producer, which publishes and feeds data to the brokers to export data to the configured consumers. The specific roles within the Kafka workflow are outlined below.

## 1.2.1 Brokers

Brokers are servers that provide the means to communicate to Kafka. Brokers manage connections to producers, consumers, topics, and replicas. A broker may be standalone or may consist of a small or large cluster of servers to form a Kafka Cluster.

The brokers that the Databridge Client for Kafka will attempt to use are specified in the client configuration file. The broker(s) identified in the configuration file are considered to be " bootstrap brokers", which are used for establishing the initial connection with Kafka while also obtaining information for the cluster. The full list of participating brokers will be obtained from the broker(s) configured.

## 1.2.2 Clusters

A Kafka cluster is a group of brokers that have been configured into an identified cluster. Brokers are grouped into clusters through a configuration file as brokers cannot be clustered without being managed.

## 1.2.3 Consumers

Consumers pull/subscribe to data in Kafka topics through brokers distribution of partitions to a consumer or a group of consumers.

## 1.2.4 Producers

The Databridge Client for Kafka acts as a producer, which "push/publish" data to brokers in the form of topics which gets exported to the configured consumers.

## 1.2.5 Topics

Topics store messages (data) that are grouped together by category.

Additionally, each topic consists of one or more partitions. When the Databridge Client for Kafka produces a message for a topic, the partition used for a particular message is based on an internal DMSII value that is unique to the DMSII record. Thus a given DMSII record will always be sent to the same partition within a topic.

The client writes JSON-formatted DMSII data to one or more topics. By default, each data set will be written to a topic that is uniquely identified by concatenating the data source name and the data set name, separated by an underscore (see Example 1-1). The configuration file parameter `default_topic` can be used to indicate that all data sets will be written to a single topic (see Example 1-2). Each JSON-formatted record contains the data source and data set name providing information regarding the source of the DMSII data if needed. Whether or not `default_topic` is used, a `topic_config.ini` file is generated in the `config` folder so each data set can be customized if so desired ( see Example 1-3).

```
; Topic configuration file for update_level 23

[topics]
backords          = TESTDB_backords
employee          = TESTDB_employee
customer          = TESTDB_customer
orddtail          = TESTDB_orddtail
products          = TESTDB_products
supplier          = TESTDB_supplier
orders            = TESTDB_orders

; Topic configuration file for update_level 23

[topics]
backords          = TestTopic
employee          = TestTopic
customer          = TestTopic
orddtail          = TestTopic
products          = TestTopic
supplier          = TestTopic
orders            = TestTopic

; Topic configuration file for update_level 23

[topics]
backords          = TestTopic
employee          = TestEmployee
customer          = TestTopic
orddtail          = TestTopic
products          = TestTopic
supplier          = TestSupplier
orders            = TestTopic
```

Additionally, each topic consists of one or more partitions. When the Databridge Client for Kafka produces a message for a topic, the partition used for a particular message is based on an internal DMSII value that is unique to the DMSII record. Thus a given DMSII record will always be sent to the same partition within a topic.

## 1.3 Data Format

The client writes replicated DMSII data to Kafka topics in JSON format. The JSON consists of several name:value pairs that correspond to data source (JSON name:value equivalent is "namespace" : data_source), data set name (JSON " name" : "dmsii_datasetname"), DMSII items (JSON " fields" : "dmsii_itemname"value pairs).

Given the DASDL:

```
EMPLOYEE DATA SET "EMPLOYEE"
POPULATION = 150
 (
EMPLOY-ID                  NUMBER(10);
LAST-NAME                     ALPHA(10);
FIRST-NAME                    ALPHA(15);
EMP-TITLE                     ALPHA(20);
BIRTHDATE                     ALPHA(8);
HIRE-DATE                     ALPHA(8);
ADDRESS                        ALPHA(40);
CITY                            ALPHA(15);
REGION                        ALPHA(10);
ZIP-CODE                      ALPHA(8);
COUNTRY                        ALPHA(12);
HOME-PHONE                    ALPHA(15);
EXTENSION                     ALPHA(5);
NOTES                          ALPHA(200);
ALPHAFILLER                   ALPHA(10);
REPORTS-TO                     NUMBER(10);
NULL-FIELD                    ALPHA(10);
CREDB-RCV-DAYS             NUMBER(5) OCCURS 12 TIMES;
ITEMH-ADJUSTMENT           NUMBER(4) OCCUTS 5 TIMES;
EMPLOY-DATE                       DATE  INITIALVALUE = 19570101;
FILLER                     SIZE(30);
 ), EXTENDED=TRUE;
```

A sample JSON record conforming to the DASDL layout above will be formatted as follows:

```
{
    "namespace":"TESTDB",
    "name":"employee",
    "fields":
    {
        "update_type":  0,
        "employ_id":111,
        "last_name":"Davolio",
        "first_name":"Nancy",
        "emp_title":"Sales Representative",
        "birthdate":"2/24/50",
        "hire_date":"6/5/87",
        "address":"507 - 20th Ave.E. Apt. 2A",
        "city":"Seattle",
        "region":"WA",
        "zip_code":"76900",
        "country":"USA",
        "home_phone":"(206)555-9856",
        "extension":"5467",
        "notes":"Education includes a BA in Psychology from Colorado State University in 1970.  She also completed ""The Art of the Cold Call.""  Nancy
is a member of Toastmasters International.",
        "alphafiller":"XXXXXXXXXX",
        "reports_to":666,
        "null_field":null,
        "credb_rcv_days_01":11,
        "credb_rcv_days_02":21,
        "credb_rcv_days_03":32,
        "credb_rcv_days_04":43,
        "credb_rcv_days_05":54,
        "credb_rcv_days_06":65,
        "credb_rcv_days_07":76,
        "credb_rcv_days_08":87,
        "credb_rcv_days_09":98,
        "credb_rcv_days_10":109,
        "credb_rcv_days_11":1110,
        "credb_rcv_days_12":1211,
        "itemh_adjustment_001":1,
        "itemh_adjustment_002":10,
        "itemh_adjustment_003":100,
        "itemh_adjustment_004":1000,
        "itemh_adjustment_005":10000,
        "employ_date":"19570101000000",
        "update_time":"20190308080314",
        "audit_ts":"20190308090313"
```

```
    }
}
```

# 2. Getting Started

The material in this chapter supplements Chapter 2 of the Databridge Client Administrator's Guide. As stated previously, the Databridge Client for Kafka uses client control tables in the same way as relational database clients. These control tables are common to all clients and have the same layout. See Databridge Client Control Tables in the Databridge Client Administrator's Guide for a complete description of these tables. All Databridge Client commands must be run with the current directory set to the working directory associated with the data source, as these commands rely on the files in the config sub-directory and write the output to the logs folder.To customize how DMSII data sets get mapped to their corresponding tables, you must write user scripts in SQL. For instructions, see Customizing with User Scripts in the Databridge Client Administrator's Guide.

- Creating Client Control Tables
- Defining a Data Source
- The Generate Command
- The Process Command
- The Clone Command

## 2.0.1 Creating Client Control Tables

Before you clone a DMSII data source, you must create the Client control tables using the Client `configure` command.

Creating Client Control Tables

1. Create the global working directory.
2. Create the data source working directory.
3. Set the current directory to be the data source working directory.
4. Create a `config` sub-directory.
5. Run an `export` command to create a text configuration file in the `config` folder.
6. Edit the text configuration file to setup the Oracle connection parameters as well as the Kafka specific parameters. Refer to the **Getting Started** chapter of the Databridge Client Administrator's Guide.
7. Run an `import` command to create a binary configuration file in the `config` folder which the Client will use by default.
8. Run a `configure` command to create the Client control tables.

The material in the **Creating Client Control Tables** chapter of the Databridge Client Administrator\'s Guide applies to the Kafka Client.

## 2.0.2 Defining a Data Source

Refer to **Defining a Data Source** chapter of the Databridge Client Administrator's Guide. All information in this chapter applies to the Kafka Client except for the data table creation and index creation user scripts as these do not apply to the Kafka Client.

The **Decoding DMSII Dates, Times, and Date/Times** section of the Databridge Client Administrator\'s Guide applies to the Kafka Client with the exception that the dates are written out using the format specified by the `span_date_format` parameter in the `[Kafka]` section of the client configuration file.

The **Creating Indexes for Tables** section of the Databridge Client Administrator\'s Guide does not apply to the Kafka Client.

The Databridge Client Administrator\'s Guide section on **Adding a Non DMSII Column** applies to the Kafka Client; in fact it supports more types of external columns than the relational database clients do. The table below shows the values for all the external column types supported by the Kafka Client.

| Bit | Value | User Column Name | Description |
|---|---|---|---|
| 1 | 1 | update_type | Database update type, as follows:0 for extract1 for create2 for delete (bit 10 must also be enabled)3 for modify<br><br>**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.<br><br>**Note**: You cannot use this bit at the 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 extract1 for create2 for delete 3 for modify<br><br>**NOTE:**<br>• If you reuse the key for this record, inserting the new duplicate record removes the key.<br>• 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.<br>• 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.<br><br>**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 | N/A |

| 19 | 262144 | user_column3 | N/A |
|----|--------|--------------|-----|
| 20 | 524288 | user_column4 | N/A |
| 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
>
> 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 are bits 14, 18, 19, and 20 which result in a dms_subtype of 0. Bits are numbered from right to left; the right-most bit is 1.

## 2.0.3 The Generate Command

The `generate` command creates a schema file for every table associated with the data sets being replicated. The files, which are named `tablename_schema.json`, reside in the dbscripts directory. This file describes the layout of the corresponding table in JSON format. An example of such a file for the table customer is below:

```
{
    "namespace": "TESTDB",
    "name": "customer",
    "pk_name": "cust_rsn_set",
    "pk_members": "(thersn)",
    "fields": {
        "update_type": "int",
        "thersn", "string",
        "acct_number", "int",
        "address", "string",
        "phone_number", "string"
    }
}
```

In addition to creating the schema files, the `generate` command also creates the `topic_config.ini` file in the config folder that allows you to associate a topic name with one or more tables. By default the client uses the topic name of `source_tablename` for the various tables. This can be changed by editing the file and setting the topic name to use the desired names. For an example of such a file for the data source TESTDB refer to example 1-1 in chapter 1.

Under normal circumstance the above mentioned files are only updated when the parent data set has bit 2 of the `status_bit` column set. Adding the -u option to the command causes all the schema files and the `topic_config.ini` file to be updated. The client always reads the `topic_config.ini` file before saving the topic names. If the file does not exist it gets created with the default topic names.

## 2.0.4 The Process Command

The `process` command is the main command of the Databridge Kafka Client. This command creates output streams for all data set tables where the active column is 1 in the corresponding entries of the DATASETS client control table. Since the `define` command initializes the ds_mode column to 0 (indicating that the data set needs to be cloned), 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.

Once the data sets are cloned, subsequent runs of the `process` command will store updates in the Kafka repository. For insert operations all columns will be present, for delete operations, only columns that are keys will be present in the update record. In the case of updates, the keys and changed columns will be present in the record. All records contain the update_type field, which is 0 for extract records, 1 for insert operations after the data extraction completes, 2 for delete operations, and 3 for update operations.

The `process` command to update the data in the Kafka repository can be scheduled. The schedule becomes effective after the `process` command is run for the first time. For more information, see the **Scheduling Updates** section in the Databridge Client Administrator's Guide.

Before running the `process` command, ensure that the current directory is set to the working directory that was created for this data source.

For more information on the `process` command refer to the Databridge Client Administrator's Guide.

## 2.0.5 The Clone Command

Refer to **The Clone Command** section in chapter 3 of the Databridge Client Administrator's Guide for documentation on this command.

# 3. Installing Databridge Client for Kafka

The Databridge Client for Kafka is installed in the same manner as the Linux Databridge Client for Oracle and requires access to an Oracle database to store metadata. It differs in that the replicated data are in Kafka topics.

# 4. Appendix

This appendix lists the configuration parameters for Databridge Kafka 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**

- Client Configuration Files
- Command-Line Options
- Syntax
- Sample Kafka Client Configuration File
- Configuring Databridge Client for Kafka Parameters
- Processing Order
- Parameter Descriptions

## 4.1 Client Configuration Files

The Databridge Kafka Client uses binary configuration files. The configuration file is named `dbridge.cfg` and resides in the `config` subdirectory of the data source\'s working directory. You can create a text version of this file using the `export` command, which can be edited and converted into a binary file using the `import` command.

## 4.2 Command-Line Options

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

| Option | dbutility Command | Description |
|---|---|---|
| ? | | Short help |
| -d | All | Default Tracing |
| -f *filename* | All | Specifies the configuration 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 |
| -t mask | All | Tracing options |
| -u | configure, define, redefine, and dropall | Unconditionally performs the requested command, overriding any warnings that would be displayed without this option. |
| -v | All | Causes the client to log and/or print some additional messages. |
| -w | clone or process | Toggles the setting of the 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 process | Instructs the client to reclone all data sets whose ds_mode has a value of 11 or 12. |
| -B | display | Causes the display command to quit after displaying the DATASETS client control table records. |
| -D | database | Specifies the Oracle database name to connect to. |
| -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. |
| -P *password* | All | Sets the password associated with the user ID for the Oracle 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 *userid* | All | Specifies the user ID for the Oracle database. The user ID must have the appropriate resource privileges for the designated relational database. |
| -X | define, redefine, clone, process | Specifies the host password. |

| | | |
|---|---|---|
| -Y reclone_all | process | Causes all active data sets to be recloned. |

## 4.3 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].

## 4.4 Sample Kafka 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 7.0 Kafka configuration file -- generated programmatically
;

[Signon]
;user                  = USERID
;password              = PASSWORD
;database              = DATABASE
;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

[Kafka]
;
;  Kafka client parameters
;
;default_topic         = ""
kafka_broker           = ""
;kafka_debug           = ""
ltrim_zeroes           = true
real_format            = scientific,11,6
rtrim_spaces           = true
security_protocol      = "PLAINTEXT"
;ssl_ca_location       = ""
;ssl_cert_location     = ""
;ssl_key_location      = ""
;ssl_key_passwd        = ""
;sasl_kerberos_srvc_name= ""
```

```
;sasl_kerberos_keytab    = ""
;sasl_kerberos_principal= ""
span_date_delim          = ""
span_date_format         = 21
span_date_scale          = 0
treat_real_as            = real
use_plus_sign            = false

[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       = true
force_aa_value_only       = 0
miser_database           = false
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       = false
;
; (2) process/clone command parameters
;
alpha_error_cutoff       = 10
automate_virtuals        = false
aux_stmts                = 100
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
null_digit_value         = 9
numeric_date_format      = 23
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_ctrl_tab_sp          = 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]
;
; (5) miscellaneous command parameters
;
display_active_only      = true
;
; (6) user scripts
;
user_script_bu_dir       = ""
user_script_dir          = "scripts/"
;
```

```
; (7) external data translation parameters
;
use_ext_translation    = false
eatran_dll_name        = "DBEATRAN.SO"

[Scheduling]
;
; dbutility process command only
;
;daily                 = 08:00, 12:00, 17:00, 24:00
;sched_delay_secs      = 600
exit_on_error          = false
sched_minwait_secs     = 0
sched_retry_secs       = 60
;blackout_period       = 00:00, 02:00

[EbcdicToAscii]
; e1 = a1
; e2 = a2
;  ...
; en = an
;
[DBConfig]
default_date_fmt       = 21
```

## 4.5 Configuring Databridge Client for Kafka Parameters

Certain Kafka specific parameters in the `dbridge.ini` must be configured before use and are outlined below. All of the Kafka-related parameters are in the `[Kafka]` section of the `dbridge.ini` file. The sample `dbridge.ini` file excerpt below shows how the Databridge Kafka Client might be configured to use Kerberos authentication.

```
[Kafka]
;
; Kafka client parameters
;
default_topic = "TESTDB"
kafka_broker = "kafkabuild.kafkalab.net:9093"
;kafka_debug = ""
ltrim_zeroes = true
real_format = scientific,11,6
rtrim_spaces = true
security_protocol = "SASL_PLAINTEXT"
;ssl_ca_location = ""
;ssl_cert_location = ""
;ssl_key_location = ""
;ssl_key_passwd = ""
sasl_kerberos_srvc_name = "kafka"
sasl_kerberos_keytab = "/etc/security/keytabs/dbclient.keytab"
sasl_kerberos_principal = "dbclient/oel.kafkalab.net"
span_date_delim = ""
span_date_format = 21
span_date_scale = 0
treat_real_as = real
use_plus_sign = false
```

Refer to the `[Kafka]` descriptions below for more information on the Kafka Client parameters.

## 4.6 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.

# 5. Parameter Descriptions

## 5.1 [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 Oracle database service name, as well as the userid/password that the Client needs to sign on to the Oracle database, where the Client control tables reside.

| Parameter | Description |
|---|---|
| hostpasswd | Default: None<br><br>Range: 17 alphanumeric characters<br><br>Command-line option: `-X`<br>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:<br><br>`DBServer`<br><br>`    KEY = "Secret"`<br><br>`dbridge.cfg`<br><br>`    hostpasswd = Secret` |
| password | Default: None<br><br>Command-line option: `-P`<br><br>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.<br><br>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:<br><br>`password = "a$bb%"`<br><br>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](NEED LINK) in the Databridge Client Administrator's Guide or see [Export the Client Configuration to a File](NEED LINK) in the Databridge Client Console Help. Passwords communicated between the Databridge Client and Databridge Server are not encoded. |
| user | Default: Login userid<br><br>Command-line option: `-u`<br><br>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. |

## 5.1.1 [signon] parameters with equivalent environment variables

| [signon] Parameter | Environment Variable | Option | dbutility Command |
|---|---|---|---|
| database | DBDATABASE | -D | All |
| hostpasswd | DBHOSTPW | -X | define, redefine, process, clone, and switchaudit |
| password | DBPASSWD | -P | All |
| user | DBUSERID | -U | All |

| [signon] Parameter | Environment Variable | Option | dbutility Command |
|---|---|---|---|

## 5.2 [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.

## 5.3 [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.

## 5.4 [Kafka]

This section contains parameters specific to Databridge Client for Kafka.

### 5.4.1 `default_topic`

**Default:**""

DMSII data set to Kafka topic mapping is controlled by the configuration file `topic_config.ini`. This file is created during the `generate` phase of the replication process. If most or all of the data sets are going to be written to a single topic, then the `dbridge.ini` file parameter `default_topic` can be used to set the topic for all of the data sets. If `default_topic` is omitted, then each data set will be assigned a topic that is comprised of the data source and data set name separated by an underscore.

### 5.4.2 `kafka_broker`

**Default:**""

The `kafka_broker` parameter consists of a comma separated list of brokers. This can be a partial list as the client will query cluster metadata information to obtain the full list of brokers and leaders.

### 5.4.3 `kafka_debug`

**Default:**""

The `kafka_debug` parameter controls the debugging contexts within the Kafka module. It consists of a comma separated list of one or more of the following: **generic**, **broker**, **topic**, **metadata**, **feature**, **queue**, **msg**, **protocol**, **cgrp**, **security**, **fetch**, **interceptor**, **plugin**, **consumer**, **admin**, **eos**, and **all**. This parameter is normally commented out and should be activated under the guidance of Micro Focus Customer Care support.

### 5.4.4 `ltrim_zeroes`

**Default:**True

The `ltrim_zeroes` parameter controls whether or not to strip leading zeroes from numbers in the output stream.

### 5.4.5 `real_format`

**Default:**scientific, 11, 6

The `real_format` parameter has 3 arguments, which specify the format type, the length, and the scale to use when formatting a DMSII REAL. 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 to the output stream.

### 5.4.6 `rtrim_spaces`

**Default:**True

The `rtrim_spaces` parameter controls whether or not to strip trailing blanks in strings written to the output stream. This parameter can be augmented by the `[Params]` section parameter `set_blanks_to_null`, which causes blank items to be represented as null.

### 5.4.7 `security_protocol`

**Default:**"PLAINTEXT"

The `security_protocol` parameter determines how the Databridge Kafka Client will communicate with the Kafka broker. This required parameter may be one of the following:

- PLAINTEXT -- no authentication or datastream encryption will be used.
- SASL_PLAINTEXT -- the client will be authenticated with Kerberos, but the datastream will not be encrypted.
- SSL -- the datastream will be encrypted but the client will not be authenticated.
- SASL_SSL -- the client will be authenticated and the datastream will be encrypted.

### 5.4.8 ssl_ca_location

**Default:**""

This parameter specifies the location of the CA certificate file to verify the broker's certificate.

### 5.4.9 ssl_cert_location

**Default:**""

This parameter specifies the location of the client's certificate PEM file.

### 5.4.10 ssl_key_location

**Default:**""

This parameter specifies the location of the client's key file.

### 5.4.11 ssl_key_passwd

**Default:**""

This parameter allows one to specify the key file password if needed. After a dbutility export, this parameter will be encoded.

### 5.4.12 sasl_kerberos_srvc_name

**Default:**""

This parameter sets the Kafka Kerberos service name. The default value is "kafka".

### 5.4.13 sasl_kerberos_keytab

**Default:**""

This parameter sets the client's keytab location. You must create the client principal and keytab file, then securely copy the keytab file to the client host. For example, using kadmin.local and the names in the sample `dbridge.ini` file below:

```
# kadmin.local -q 'addprinc -randkey dbclient/oel.kafkalab.net@KAFKALAB.NET'
# kadmin.local -q 'ktadd -k /etc/security/keytabs/dbclient.keytab dbclient/oel.kafkalab.net@KAFKALAB.NET
```

### 5.4.14 sasl_kerberos_principal

**Default:**""

This parameter sets the client principal name. For example, from the sample `dbridge.ini` above: dbclient/oel.kafkalab.net

### 5.4.15 span_date_delim

**Default:**\"\"

The `span_date_delim` parameter allows the year, month, and day to be separated either 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.

## 5.4.16 span_date_format

**Default:**21

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.

## 5.4.17 span_date_scale

**Default:**0

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.

## 5.4.18 treat_real_as

**Default:**real

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.

## 5.4.19 use_plus_sign

**Default:**False

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

## 5.5 [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 Databridge Client for Kafka. 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.

### 5.5.1 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.<br><br>NOTE:The brackets and value are require 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 or 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

## 5.5.2 Process and Clone Command Parameters

**Process and Clone Command Unchanged Parameters**

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

## 5.5.3 Server Option Parameters

See Appendix C of the Client Administrator's Guide.

## 5.5.4 Display Command Parameters

See Appendix C of the Client Administrator's Guide.

## 5.5.5 User Scripts Parameters

See Appendix C of the Client Administrator's Guide.

## 5.6 [Scheduling]

See Appendix C of the Client Administrator's Guide.

## 5.7 [EbcdictoAscii]

See Appendix C of the Client Administrator's Guide.

## 5.8 External Data Translation DLL Support

The Kafka Client does not currently use a data translation DLL.

# 6. 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.

**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 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.

**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.

**Kafka Specific Terms**

The following terms are used to describe certain terms that are specifically related to Databridge Client for Kafka.

- **Brokers.** Brokers may consist of a small or large cluster of servers running Kafka.
- **Bootstrap Brokers.** Brokers that enable a cluster of servers to be accessible through the initial connection.
- **Producers.** Producers push and/or publish data to Kafka topics which are managed within a cluster of servers managing topics (Brokers).
- **Topics.** Topics are where data (messages) is stored and published to Broker(s). Topics can be published to multiple brokers, and the data can be distributed in different ways through the use of partitions.

**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.

**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.

**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.

**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.

# 7. Legal Notice