



# Errors and Messages Guide

## Databridge

version 6.6 Service Pack 1

## **Legal Notices**

© Copyright 2020 Micro Focus or one of its affiliates.

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.

### **Patents**

This Micro Focus software is protected by the following U.S. patents: 6983315, 7571180, 7836493, 8332489, and 8214884

### **Trademarks**

Micro Focus, the Micro Focus logo, and Reflection among others, are trademarks or registered trademarks of Micro Focus or its subsidiaries or affiliated companies in the United Kingdom, United States and other countries. RSA Secured and the RSA Secured logo are registered trademark of RSA Security Inc. All other trademarks, trade names, or company names referenced herein are used for identification only and are the property of their respective owners.

### **Third-Party Notices**

Third-party notices, including copyrights and software license texts, can be found in a 'thirdpartynotices' file located in the root directory of the software.

---

# Contents

<b>About This Guide</b>	<b>5</b>
<b>1 Databridge Host and Enterprise Server Messages</b>	<b>7</b>
0000-0899 (DBEngine)	7
0900-0999 (Accessories)	23
1000-1099 (DBServer)	25
1100-1199 (DBEnterprise)	28
1200-1299 (DBSpan)	36
1300-1399 (DBSnapshot)	38
1400-1499 (DBSupport)	38
1500-1599 (DBAuditTimer)	39
1600-1699 (DBGGenFormat)	39
1900-1999 (DBTwin)	43
4000-4099 (DBDMSIIClient)	44
Status Messages	46
Windows Event DBDirector Messages	49
<b>2 Databridge Client Messages</b>	<b>51</b>
Command Messages	51
Common Log Messages	52
Configure Command Messages	55
Define Command Messages	56
Redefine Command Messages	58
Generate Command Messages	64
Reorg Command Messages	65
Process and Clone Commands Messages	66
Drop and Dropall Commands Messages	84
Switchaudit Command Messages	85
Display Command Messages	85
Runscript Command Messages	86
Unload Command Messages	86
Reload Command Messages	87
Refresh Command Messages	87
Export Command Messages	88
Import Command Messages	88
Rowcounts Command Messages	88
CreateScripts Command Messages	89
Tcptest Command Messages	90
Databridge Client Console Messages	90
Client Errors	96
Usage Errors	149
Host Messages That Appear on the Client	150
Sockets Errors	155
ODBC Errors	156
OCI Errors	157

Client Warnings .....	158
Messages Associated with Client Exit Codes .....	186
<b>3 Databridge Client Manager Service/Daemon Messages</b>	<b>189</b>
Log Messages .....	189
Error Messages .....	195
Windows Application Log Messages .....	203
Client Console and Configurator Messages .....	205

# About This Guide

This guide provides detailed descriptions of the error, status, and log messages that may occur when using Databridge.

To quickly locate a message in this PDF document, use the **Find** function (Ctrl + F) to search by numbers or unique keywords in the message.

For troubleshooting assistance, see the Administrator's Guide for the product in question.

## Conventions

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

<b>This convention</b>	<b>Is used to indicate this</b>
<code>this type style</code>	text that you type, filenames and directory names, onscreen messages
<i>italic</i>	variables (for example a file name or a data source name that appears in a message)
square brackets ( [ ] )	part of a message that may not always be present
pipe (   )	part of a message this can have more than one form. When enclosed in braces ( { } ), one of two forms will always be present.



# 1 Databridge Host and Enterprise Server Messages

This section lists error messages generated by the Databridge Host components and Databridge Enterprise Server in error number order.

Messages may appear on the station running a component linked to DBEngine and in the component's report file. On the MCP server, messages are displayed by the default error handler in DBSupport. Custom error handling routines can help you manage error-prone situations by performing an action when a particular error is encountered. For example, if Databridge Server encounters an error while trying to notify a client system, an error handling routine may cause Databridge Server to retry the notification. For information about creating custom error handling routines, see the *Databridge Programmer's Reference*.

## In this Chapter

- ♦ ["0000-0899 \(DBEngine\)" on page 7](#)
- ♦ ["0900-0999 \(Accessories\)" on page 23](#)
- ♦ ["1000-1099 \(DBServer\)" on page 25](#)
- ♦ ["1100-1199 \(DBEnterprise\)" on page 28](#)
- ♦ ["1200-1299 \(DBSpan\)" on page 36](#)
- ♦ ["1300-1399 \(DBSnapshot\)" on page 38](#)
- ♦ ["1400-1499 \(DBSupport\)" on page 38](#)
- ♦ ["1500-1599 \(DBAuditTimer\)" on page 39](#)
- ♦ ["1600-1699 \(DBGenFormat\)" on page 39](#)
- ♦ ["1900-1999 \(DBTwin\)" on page 43](#)
- ♦ ["4000-4099 \(DBDMSIClient\)" on page 44](#)
- ♦ ["Status Messages" on page 46](#)
- ♦ ["Windows Event DBDirector Messages" on page 49](#)

## 0000-0899 (DBEngine)

0000 OK

This is a notification that the last call to a DBEngine entry point was completed without error.

0001 Commit transaction group

This is a notification that DBEngine encountered the end of a transaction group. The accessory should now commit all updates it has received for this group.

0002 Abort transaction group

DBEngine has encountered an indication that the current transaction group was aborted. The accessory should discard all updates it has received in this transaction group.

0003 Caller canceled

An accessory's callback procedure returned a FALSE, which indicates that it does not want to receive any more callbacks.

0004 Invalid database name: *name*

An accessory passed an invalid database name to an entry point. Check the input to the accessory and try again.

0005 Invalid dataset name: *name*

An accessory passed an invalid data set name to an entry point. Check the input to the accessory and try again.

0006 Invalid data item name: *name*

An accessory passed an invalid data item name to an entry point. Check the input to the accessory and try again.

0007 *message*

where *message* is one of the following messages:

"Active (*usercode*)database audit file *afn* is unavailable"

The audit file is still in use by the Access routines. This is normal and indicates that the accessory should stop.

"Active (*usercode*)database audit file *afn* has no more audit available"

Databridge Plus has gathered all of the updates from the current audit file up to the last QPT.

"(*usercode*)database Audit file *afn* is an incorrect audit file number"

DBEngine was given an AFN less than one or greater than 9999. Check the values that the accessory passes to DBEngine to make sure they're not invalid.

"Another program opened (*usercode*)database audit file *afn* EXCLUSIVE"

A different program has opened the audit file with the EXCLUSIVE attribute set to TRUE. Rerun the accessory when that program has finished.

"(*usercode*)database Audit file *afn* was not found"

An operator entered a *mixnumber* NF command to indicate that the audit file is not available. Reload the audit file and run the accessory again.

"(*usercode*) database Audit file *afn* missing section"

Reload the missing sections of the audit file and run the accessory again.

"(*usercode*)database Audit file *afn* is offline"

The audit file is archived offline. Retrieve the audit file, load it, and run the accessory again.

"(*usercode*)database Audit file *afn* is unavailable (no family *packname*)"

The pack that contains the audit file is unavailable. Make the pack ready and rerun the accessory.



"(usercode)database Audit file *afn* does not exist because the database was never updated."

The database has never been updated by an application program and no audit files exist. Run an application program that updates the database and then rerun the accessory.

0008 Missing DESCRIPTION file for Audit file update level *updatelevel*

DBEngine cannot find a DMSII DESCRIPTION file that is compatible with the audit file. *updatelevel* gives the DMSII update level for the audit file. Copy the DESCRIPTION file that has the compatible update level from one of the following backups and rerun the accessory:

DESCRIPTION/*database*/updatelevel

*updatelevel*/DESCRIPTION/*database*

This message can also appear when creating the secondary database if the primary database was dumped before an update program was run following the DASDL update to add the DBTWINCONTROL data set. If so, complete the procedure "Cloning Procedure on the Primary System," in the *Databridge Twin Administrator's Guide*, starting at step 2.

Additionally, make sure that you copied the primary database DESCRIPTION file to the secondary side before compiling the (modified) secondary database DASDL. If you didn't, copy the DESCRIPTION file and complete "Installing Databridge Twin to the Secondary System," starting at step 6.

0009 Audit file *afn*: Unexpected end-of-file

DBEngine encountered an unexpected end-of-file error while reading the audit file given by *afn*. Most likely the audit file is corrupted. Use COPYAUDIT to reload the audit file from a backup and then rerun the accessory.

0010 Data set *structure* not selected

An accessory requested information for a data set that it had not previously selected via the DBSELECT entry point. *structure* is the structure number of the data set that was not previously selected. This typically indicates a problem in a user-written accessory. Check the user-written source code and try again.

0011 Invalid audit location for *dataset*; *value\_2* must be from *value\_3* to *value\_4*

An accessory tried to select a data set but gave an invalid audit location:

*dataset* indicates the data set

*value\_2* is the name of the audit location component that is invalid

*value\_3* and *value\_4* give the legal range for that component

Check the input to the accessory. If the accessory is Span, check the Span parameter file to make sure that all of the audit locations have legal values. Correct any illegal values and try again.

0012 Invalid dataset structure number: *structure*

An accessory passed an invalid data set structure number to an entry point. *structurenumber* in this case is not a data set in the DMSII database. Instead, it could be one of the following:

The structure number for a set. In this case, check the input to the accessory and correct the structure number.

—or—

If the accessory specified a logical database, it could be a data set that exists in the physical database but not in the logical database. In this case, select a different logical or physical database.

0013 Out of room in array *arrayname*

DBEngine was unable to allocate more memory for one of its internal arrays. If this error occurs consistently, report this error to Technical Support.

0014 Invalid attribute number *attrnum*

The accessory requested file attribute information but gave an invalid file attribute number.

0019 No datasets or remaps selected

An accessory requested updates from DBEngine, but it did not previously select any data sets or remaps.

0020 Table reorganization required for *datasetname*; DMS dataset level = *updatelevel*, Table level = *clienttablelevel*

The DMSII update (format) level for *datasetname* does not match the update level for the corresponding client table. After making any necessary changes to the table in the client database, update the client table level information to match the DMSII update level for *datasetname*. This message applies to any accessory that calls the DBREAD entry point in DBEngine.

0021 Table purge required for *clienttablename*

The audit trail contains a record indicating a selected data set was initialized (using DMUTILITY). Purge the table in the client database and then set the mode value back to normal (2) for that table. This message applies to any accessory that calls the DBREAD entry point in DBEngine.

0022 Database timestamp mismatch: Audit file has *timestamp*, DESCRIPTION file has *timestamp*

The database timestamp in the audit file does not match the database timestamp in the DESCRIPTION file. Find the correct DESCRIPTION file and make it available to Databridge.

0023 Array is too short: *arrayname*

An accessory passed an array as a parameter to one of the DBEngine entry points, but the array was shorter than DBEngine expected. Typically, this error occurs when an accessory is not recompiled with the current version of the Databridge Interface. Recompile the accessory with the current Databridge interface and try again.

0024 *filetitle* is not available

DBEngine could not locate the database CONTROL file. Locate the DMSII database CONTROL file and make it available to Databridge. Make sure you check security access restrictions.

0025 *databasename* is not an audited database

The requested database is not an audited database. Databridge can use only audited databases.

0026 *modevalue* is an invalid mode for *datasetname*

An accessory tried to select a data set with an invalid mode value. (The mode value indicates whether the data set should be cloned, processed normally, etc.) In this case, set the mode to the correct value, as follows:

Mode = 0	The data set needs to be cloned.
Mode = 1	The data set is in the fix up phase
Mode = 2	The data set is in the update phase
Mode = 3	The data set was reorganized.
Mode = 4	The data set was purged

This message applies to any accessory that calls the DBSELECT entry point in DBEngine.

**NOTE:** Mode values also apply to the Databridge Clients. In the *Databridge Client's Administrator's Guide*, look for `ds_mode` in the client control tables.

0027 *datasetname* does not have a set with a unique key

An accessory tried to initiate cloning for the data set (indicated by *datasetname*) which does not have a NO DUPLICATES set. Cloning of certain types of data sets requires a NO DUPLICATES set. One option is to define a primary key for the data set.

0028 Databridge not initialized

The accessory called an entry point in DBEngine that requires a previous (successful) call of DBINITIALIZE, which determines the database being replicated. Check your user-written accessory to make sure you initialize DBEngine before you call any other entry points.

0029 DMS Error: *subtype*

*subtype* contains the text of a DMS error message. Consult your Unisys DMSII manuals for more information on this message.

0030 Invalid set structure number: *number*

An accessory passed an invalid set structure number to an entry point. *number* in this case is not a set in the DMSII database. Instead, it could be one of the following:

The structure number for a data set. In this case, check the input to the accessory and correct the structure number.

—or—

If the accessory specified a logical database, it could be a set that exists in the physical database but not in the logical database. In this case, select a different logical or physical database.

0031 Invalid data item number: *itemnumber* not found in dataset

The accessory passed an invalid data item number to an entry point. The data set does not have a data item with that number. Most likely, the accessory passed the wrong data set and/or item number to the DBNULL entry point. Correct the user-written accessory. Note that the DBLAYOUT entry point returns information about individual data items, including their item numbers.

0032 Invalid variable format record type number for dataset #*structurenumber*

The accessory passed an invalid record type number for a variable format data set. Check your user-written accessory for the variable format record type number. The DBDATASETS and DBDATASETINFO entry points return information about valid record types.

0033 *tablename*: Audit location mismatch, *subtype = value* is wrong. Check for DMS rollback.

*tablename* is the accessory-supplied name, which defaults to the data set name.

*subtype* indicates which component of the audit location is incorrect, for example, the ABSN or INX.

*value* contains the value of that component.

This error could indicate that the DMSII database was rolled back. If so, roll back the client database as well. This message applies to any accessory that uses the DBSELECT entry point. This error can also occur if you attempt a roll back but have an invalid timestamp.

0034 Internal error: Invalid node in DESCRIPTION file

DBEngine encountered an error reading the DESCRIPTION file. If this error occurs consistently, report this error to Technical Support.

0035 Internal error: Invalid STR index in *procedurename*

DBEngine generated an invalid structure index. If this error occurs consistently, report this error to Technical Support.

0036 Embedded dataset not supported: #*structurenumber structurename*

An accessory tried to select an embedded data set for processing. Databridge does not support certain types of embedded data sets at this time. In order to clone an ORDERED or COMPACT data set, NO DUPLICATES must be set.

*structurenumber* contains the structure number of the embedded data set.

*structurename* contains the embedded data set name.

0037 Database does not have a global record

An accessory tried to select the global data set for processing but the database does not have a global data set.

0038 Set *setname* does not have any keys

An accessory requested the keys for a set but the set does not have a key. *setname* contains the set name. This is an informational message.

0039 Invalid structure index: *number*

An accessory requested the name of a structure but gave an invalid number for the structure index. (The structure index is not the same as the structure number. It is the unique number returned by DBSELECT.) In this case, correct the user-written accessory to request a valid number for the structure index.

0040 Database name *databasename* not found

An accessory requested a database, but DBEngine could not find the database indicated by *databasename*. In this case, check the following:

Spelling of *databasename*

Usercode for Databridge and database

Usercode specified by CONTROL file and DASDL

0041 Access to *databasename* denied until *day @ time*  
Access to *databasename* denied by operator QUIT  
Access to *databasename* denied by guardfile

*databasename* indicates the logical or physical database name. This error can occur in any of the following cases:

The DBEngine parameter file limits database access to certain hours of the day and the current time is outside of that time period

The host operator stopped the DBEngine by issuing the AX QUIT command

The database guardfile does not allow the accessory to access the database

For more information about setting up guardfiles, see the *Databridge Installation Guide*.

0042 Missing *DMSUPPORTlibrary* for update level *updatelevel*

DBEngine could not locate the DMSUPPORT library that corresponds to the indicated update level. DBEngine first reads the DMSUPPORT file title specified in the DASDL. The default is *DMSUPPORT/databasename*. It tries to link to DMSUPPORT using the following titles in order until it finds the one with the correct update level and update timestamp:

```
normalname/updatelevel  
updatelevel/normalname  
normalname
```

The error indicates that these files do not exist or they have the wrong update level and/or timestamp. To resolve this, find the correct DMSUPPORT library and copy it as *normalname/updatelevel* or *updatelevel/normalname* where the DASDL says DMSUPPORT is located.

0043 Parent of *datasetname* must be selected

*datasetname* indicates the name of an embedded data set. This occurs when an accessory tries to use the DBSELECT entry point without first selecting the parent of the embedded data set. To resolve this error, correct the user-written accessory to use DBSELECT for the parent data set first. In the Span, make sure the parameter file entries for parents precede the corresponding entries for embedded data sets.

0044 DBTWINCONTROL data items are incorrect

The DBTWINCONTROL data set for Databridge Twin Client must have a certain layout of data items. This error indicates that the data set does not have the proper layout. In this case, check the file *PATCH/DATABRIDGE/TWIN/DASDL* and make sure you inserted it exactly as is into both the primary database and secondary database DASDLs. See the *Databridge Twin Client Administrator's Guide* for more information.

0045 Error reading file: *filename*

This error indicates an I/O error on a file. *filename* is the name of the file. Possible reasons are as follows:

If this is a disk file and the error persists, the file may be corrupted. Reload it from a backup and run the accessory again.

If this is COMMPORT, there is a possibility that a network channel was broken. Try rerunning the accessory.

0047 Database must have the INDEPENDENTTRANS option set

This message applies to Databridge Twin. Modify the DASDL to set the INDEPENDENTTRANS option and then recompile the DASDL. Databridge Twin supports databases only when the INDEPENDENTTRANS and AUDIT options are set.

0048 Missing audit location information. Please run DBTwin with the LOAD or LOCATION command.

This message applies to Databridge Twin and indicates that there are no records in the DBTWINCONTROL (restart) data set. Before you attempt to run Databridge Twin in normal tracking mode, you must do one of the following:

Run Databridge Twin with the LOAD command (which populates the secondary database)

—or—

Run Databridge Twin with the LOCATION command (which provides a specified audit location in the client database as the starting point for tracking when you clone the database using a method other than the LOAD command). For more information, see "Cloning Procedure on the Secondary System" in the *Databridge Twin Administrator's Guide*.

0049 Missing Database-Stack-Terminate record at end of *afn*.

This message indicates that Databridge Twin could not find the DBST (Database-Stack-Terminate) record that marks the end of the loaded dump. *afn* is the number of the last audit file it tried. Remove the secondary database and all of its audit files and start Databridge Twin with the LOAD command again.

0050 Checksum error on Audit file *afn* ABSN *absn*

DBEngine has detected a checksum error in the audit file number *afn* at audit block serial number *absn*. This indicates that the audit file is corrupted. To resolve this, use COPYAUDIT to reload a valid copy of the audit file from backup.

0051 Internal error: Structure *structurenumber* not found in SIB

DBEngine could not find the structure indicated by *structurenumber* in the Structure Information Block (SIB) of the database. This is an internal error in DBEngine. Report this to Technical Support.

0052 DESCRIPTION file property level is too old

The database DESCRIPTION file you are trying to read was generated with an old version of DMSII. You must recompile the database with more current DMSII software before you can use Databridge.

0053 Audit file format level is too old

The database audit file Databridge is trying to read was generated with an old version of DMSII. You must recompile the database with more current DMSII software before you can use Databridge.

0054 DESCRIPTION file property level is *value\_1* but Databridge is *value\_2*

The database DESCRIPTION file you are trying to use was generated with a newer version of DMSII than Databridge was compiled for. Databridge might continue to work properly or it might not. Contact Technical Support for a newer version of Databridge.

0055 DESCRIPTION file was compiled with DASDL *n.n*

This message is displayed with the previous message (0054) for diagnostic purposes. It indicates the SSR of the DASDL compiler that built the DESCRIPTION file.

0056 Databridge is not qualified on this DMSII release

Databridge has not been tested using the version of DMSII you have. Contact Technical Support for a newer version of Databridge.

0057 Internal error: *procedurename* reference not initialized

This is an internal error in the DBEngine. Report this to Technical Support.

0058 DMUTILITY failed

Databridge Twin tried to run DMUTILITY to initialize a data set but DMUTILITY failed. Contact the database administrator to determine the cause.

0059 Repositioning tape from word *value\_1* to word *value\_2*

This is an informational message that indicates DBEngine is rewinding the audit tape to an earlier block. On some tape drives, for example, compressing drives, this would normally cause errors because they cannot read backwards or position to an earlier block. DBEngine will rewind the tape and attempt to read forward to the block you want.

0060 Bad current block size in Audit *afn* and SEG *seg*

DBEngine attempted to read the next audit block but found an invalid size for the current audit block. This could be the result of a corrupted audit file or an accessory passing an invalid audit location to DBEngine when calling DBSELECT.

*afn* is audit file number.

*seg* is segment number.

To resolve this message, try the following:

If the audit file is corrupted, reload it from backup using COPYAUDIT.

If an accessory passed DBEngine an incorrect audit location, correct the location and rerun the accessory.

0061 Audit file *afn* format level is *AFLevel* but Databridge is *DBLevel*

The database audit file you are trying to use was generated with a newer version of DMSII than Databridge was compiled for. Databridge might continue to work properly or it might not. Contact Technical Support for a newer version of Databridge.

*afn* is audit file number.

*AFLevel* is audit file format level number.

*DBLevel* is audit file format level when Databridge was compiled.

0062 Audit file was generated with DMSII *n.n*

This message is displayed with message 0061 for diagnostic purposes. It indicates the SSR of the Accessroutines that wrote the audit file.

- 0063 Extracting for *datasetname*
- DBEngine has started reading records in *datasetname*. This is an informational message.
- 0064 Reading fixup records
- DBEngine completed reading records from the database and is now applying the audit records that were generated during the extract phase. This is an informational message only.
- 0065 Finished reading fixup records
- DBEngine completed reading the audit records generated during the extract phase. This is an informational message.
- 0066 Audit location: AFN = *afn* ABSN = *absn* (*num* %)
- This message appears in response to a Databridge Twin AX STATUS command. It provides Databridge Twin's location in the primary database audit file.
- afn* is audit file number.
- absn* is audit block serial number.
- num* is percentage of audit file processed.
- 0067 Audit time: *timestamp*
- This message appears in response to a Databridge Twin AX STATUS command and displays the audit location via the *timestamp* value.
- 0068 *subtype* Create: *value\_1* Modify: *value\_2* Delete: *value\_3*
- This message appears in response to a Databridge Twin AX STATUS command to Databridge Twin. It indicates the number of records replicated to the client database.
- subtype* indicates whether these counts are for successful updates or for errors.
- value\_1* is the number of new records.
- value\_2* is the number of updated records.
- value\_3* is the number of deleted records.
- 0069 Waiting for AFN *afn* to become available
- DBEngine is waiting for an audit file to become available. Typically this occurs when DBEngine attempts to read an audit file that is in use by the Access routines. This message may appear in response to a Databridge Twin AX STATUS command.
- 0070 Terminating at end of transaction group
- This message indicates that the accessory will stop at the end of the current transaction group. This is in response to an operator request to terminate the accessory (for example, *mixnumber* AX QUIT to terminate Span).
- 0071 Started *timestamp*
- timestamp* indicates when DBEngine started running.
- 0072 Ended *timestamp*
- timestamp* indicates when DBEngine finished running.
- 0073 Version *version* compiled *timestamp*



*version* indicates the release and patch level of DBEngine and *timestamp* indicates when DBEngine was compiled.

0074 Filter *filtername* not found in *libraryname*

The filter specified by *filtername* does not exist in the library specified by *libraryname*. Check the spelling of each and check the library to make sure it actually contains the filter.

0075 *datasetname* does not have a set

Databridge Twin requires a set for each data set that is replicated. It continues, however, replicating all of the other data sets that do have sets. If you want Databridge Twin to replicate *datasetname*, add a set to that data set in both the primary and secondary databases. Or, use the `FIND` command in the Databridge Twin parameter file if the data set has suitable subsets.

0076 Entry point not supported in TwinEngine. You must have the complete version.

To use this entry point, you must have the DBEngine that is part of the complete Databridge host release. The accessory you are using is linking to the DBTwinEngine, which is part of the Databridge Twin release and does not support all of the entry points in the standard DBEngine.

0077 Guardfile missing: *guardfiletitle*

Check your guardfiles, as explained in the *Databridge Installation Guide*.

0078 Unknown NULL type: *value*

Databridge does not recognize the NULL type displayed for value. This is an internal error; if it occurs consistently, report it to Technical Support.

0079 Invalid structure number: *structurenumber*

The structure number you requested is invalid or does not exist. Check your entry and check the DASDL for the correct structure number. You can also use the Lister report to determine the correct structure number.

0080 Invalid year: *year*

The year you entered is invalid. Check your entry and the format specified in the parameter file and try again. This message applies to any program (for example, Snapshot, Span, and Databridge Twin) that uses timestamps.

0081 Invalid month: *month*

The value you entered for months is invalid. Check your entry and the format specified in the parameter file and try again. This message applies to any program (for example, Snapshot, Span, and Databridge Twin) that uses timestamps.

0082 Invalid day: *day*

The value you entered for days is invalid. Check your entry and the format specified in the parameter file and try again. This message applies to any program (for example, Snapshot, Span, and Databridge Twin) that uses timestamps.

0083 Invalid hour: *hour*

The value you entered for hours is invalid. Check your entry and the format specified in the parameter file and try again. This message applies to any program (for example, Snapshot, Span, and Databridge Twin) that uses timestamps.

0084 Invalid minute: *minute*

The value you entered for minutes is invalid. Check your entry and the format specified in the parameter file and try again. This message applies to any program (for example, Snapshot, Span, and Databridge Twin) that uses timestamps.

0085 Invalid second: *second*

The value you entered for seconds is invalid. Check your entry and the format specified in the parameter file and try again. This message applies to any program (for example, Snapshot, Span, and Databridge Twin) that uses timestamps.

0086 Compile of *supportname* failed

An accessory (for example, Span) tried to recompile a tailored Support library because a filter or format had the wrong update level. The compile failed, however, so you must compile the Support library. After the library is successfully compiled, rerun the accessory.

0087 DESCRIPTION file *descriptionfilename* is unavailable; Result code: *subtype*

Check the usercode and pack where the DESCRIPTION file resides and make sure that DBEngine has access to that usercode and pack.

0088 Starting *online\_or\_offline* extract

This status message indicates that DBEngine is starting an online or offline extract. This message applies to all Accessories that clone data.

0089 Forcing syncpoint

This status message indicates that DBEngine is forcing a syncpoint to mark the boundary of the fixup phase. If there is a significant delay between this message and a subsequent message, most likely some other program is staying in transaction state too long and preventing the syncpoint from occurring. Correct the other program so that it exits transaction state sooner.

0090 Invalid Property level in parameter file: *filename*

This message indicates that you have entered an invalid property level in the DBEngine parameter file. The property level must be six digits.

0091 Trouble formatting record at AA=*address* *auditlocation*

This message indicates a problem in a formatting routine. Specifically, it can occur when the accessory is using BINARYFORMAT and filtering null records (via the DBFilteredNullRecord API). If this message persists, notify Technical Support.

0092 Expected ABSN=*absn1* in AUDITafn at segment *seg* but found ABSN=*absn2*

This message indicates discontinuity or corruption in the audit trail, possibly resulting from an unsuccessful repositioning of the accessory after a rollback. Do the following:

Load the backup of the audit file (indicated by AUDITafn) on the primary system.

Run the accessory again. If this message no longer occurs, continue as usual. If this message reappears after you have tried the backup audit file, contact Technical Support.

0093 Stopping at task *taskname*

This status message indicates that Span, DBServer, or Databridge Twin is stopping at the indicated *taskname*, based on the task specified for the STOP option. For more information, see the STOP option for the appropriate component.

0094 Exceeded audit timestamp limit: *timestamp*

This status message indicates that the Span, DBServer, or Databridge Twin is stopping at the indicated timestamp, based on the time specified for the *STOP* option. For a list of references, see the preceding error message.

0095 Partitioned dataset *datasetname* is not supported

This release of Databridge does not support partitioned data sets. If you are using Snapshot, comment the data set name in the list of data sets. If you are using Span, comment the data set replication status information. If you are using DBServer, turn off cloning for the data set in the client control tables.

0096 DBSupport needs to be recompiled

Basically you must recompile DBSupport anytime the DMSII database update level changes. Typically the database update level changes when you reorganize the database. When Span and DBServer detect an update level mismatch, they attempt to recompile the support library automatically. If the recompile fails, this message is displayed. You must then recompile the support library manually. To recompile the support library, follow the instructions in the *Databridge Programmer's Reference*.

If you use Databridge Clients older than version 2.7, this error can occur even after DBSupport is recompiled. Either upgrade you Databridge Client to the current version or reclone all of the data set tables.

0097 *location* must be from *nnnn* to *mmmm*

*location* represents a component of an audit location, as follows:

AFN	(audit file number)
ABSN	(audit block serial number)
INX	(index)
SEG	(segment)

In this case, an accessory has passed an invalid audit location to DBEngine (during the DBSelect API). Check that the accessory parameter file or client control table contains a value that falls within the range specified by *nnnn* to *mmmm*.

0098 *subsetname* is not a set of *datasetname*

This message occurs when the (DBTwin or the DMSII Client) parameter file contains a "FIND *dataset* USING *subsetlist*" option. One of the names in the subset list is not a set or subset of the data set. Correct the subset name in the parameter file and rerun DBTwin or the DMSII Client again.

0099 *entrypoint* does not allow re-entry

This message can occur when you are using a user-written accessory. It indicates that a DBEngine entry point cannot be called recursively (directly or indirectly). For example, this message could occur if an accessory called the DBDATASETS entry point and passed it a callback routine that in turn called DBDATASETS. When this message occurs, rewrite your accessory so that it does not call entry points recursively.

0100 Incorrect ABSN in block after AFN=*afn* ABSN= *absn*

This message indicates a problem in the DMSII audit trail. Each audit block should have an ABSN that is one greater than the previous block. When this message occurs, however, DBEngine has encountered an audit block where this is not true. The most common cause of this problem is failing to remove obsolete audit files after performing a database rollback.

0101 Audit file *afn* corrupted

This message indicates that the file named `AUDITafn` is corrupted and therefore Databridge cannot read it. To remedy this situation, you may need to rebuild to a previous, uncorrupted audit file.

0102 Incorrect timestamp in audit file `AUDITafn` near `ABSN absn`

This message indicates a problem in the DMSII audit trail. Each audit block should have its own timestamp and the timestamp of the previous audit block. When this message occurs, however, DBEngine has encountered an audit block where this is not true. The most common cause of this problem is failing to remove obsolete audit files after performing a database rollback.

0103 Statistics are not available. Compile Databridge Engine with `$ SET STATS`

This message indicates an accessory has called the `DBStatistics` entry point but the DBEngine was not compiled with the `$ STATS` option. Load the DBEngine code file that was compiled with the `$ STATS` option and then rerun the accessory.

0104 Filter excluded record

This message indicates the filter rejected the record, that is, the record did not satisfy the filter. Normally, this message is not visible. It is used internally to communicate between the filter and the accessory.

0105 Database update level changed from *previousupdatelevel* to *updatelevel*

This message indicates that the DMSII database update level of the primary system does not match the update level of the secondary system. You must reclone the primary database.

0106 Invalid timestamp: *nextimestamp*

This message indicates that `DBSPLITTIMESTAMP` detects an invalid timestamp. Correct the specified invalid timestamp.

0107 Invalid processing parameter type *nnn*

This message indicates that the accessory passed an invalid processing parameter type to the `DBParameters` entry point. Correct the accessory and run it again. See the list of valid parameter types in `SYMBOL/DATABRIDGE/INTERFACE` below the comment line "`DBPARAMETERS processing parameter types.`"

0108 Restart dataset is unsuitable for `DBTwin`

This message indicates that the restart data set does not meet the required criteria. Either modify the layout of the restart data set or add the `DBTWINCONTROL` data set to your primary database `DASDL` source. Refer to the *Databridge Twin Administrator's Guide* for more information.

0109 Audit switch wait exceeded *maxdelay* seconds

This message indicates that the `DBAUDITSWITCH` has failed and has exceeded the maximum delay retry rate. This problem could also be caused by some other application program remaining in transaction state too long.

0110 Invalid audit medium value: *nnnn*

This message indicates that the `AUDIT ON` parameter in either the Span, Server, or parameter file is invalid. Enter a valid `AUDIT ON` parameter into the appropriate parameter file. See the list of valid audit mediums in `SYMBOL/DATABRIDGE/INTERFACE` below the comment line "`DBAuditMedium parameters.`"

- 0111 Invalid audit type value: *nnnn*
- This message indicates that an accessory called the DBAuditMedium entry point in DBEngine with an invalid audit type. Correct the audit type value. See the list of valid audit types in SYMBOL/DATABRIDGE/INTERFACE below the comment line "DBAuditMedium parameters".
- 0112 Port *portnumber* at server is unavailable *reasoncode*.
- This message indicates that the specified port does not exist or is unavailable. Verify that the SOURCE entry is correct in the accessory's parameter file. The *reasoncode* is a number indicating the type of open error.
- 0113 Can't track *datasetname* without INDEPENDENTTRANS
- This message indicates that the mode of the embedded data set *datasetname* is one (fixup) or two (normal) and the database does not have the INDEPENDENTTRANS option set. The embedded data set must have a mode of zero.
- 0114 Program fault
- This message indicates that an internal error occurred. Contact Technical Support.
- 0115 Interface version mismatch: Databridge Engine = *engineversion*, accessory: *accessoryversion*
- This message indicates that DBEngine and the interface of the specified accessory are incompatible. Recompile the accessory using the current Databridge interface file and retry the task.
- 0116 Unknown filter name: *filtername*
- DBSupport returns this message if the accessory specifies an unknown filter name. Verify the filter names in the accessory's parameter file.
- 0117 Invalid AFN in *DMSIIcontrolfilename*: *afn*
- This message indicates that the AFN in the DMSII CONTROL file is invalid. One possible reason for this error is that the database administrator did a DMCONTROL INITIALIZE on the DMSII CONTROL file and no update programs have been run since then.
- 0118 Invalid task name: *taskname*
- This message indicates that an accessory supplied an invalid program name to the DBLIMITTASKNAME entry point.
- 0119 Recovery point not found
- This message indicates that the Engine could not find the Rollback or Rebuild point in the audit trail. Consult the Rollback or Rebuild DMSII reports to determine the recovery point.
- 0120 Database rolled back to *AFN=afn ABSN=absn Seg=seg Inx=inx timestamp*
- This message indicates that DBEngine detected that an accessory supplied an invalid audit location. DBEngine will automatically look for a database rollback (or rebuild) and use it. The accessory should reload its client database as of that point before trying to reprocess. Use the timestamp displayed in this message for the timestamp to specify in the DBSelect call.
- 0121 Missing transform *transformname*
- DBSupport returns this message if the accessory specifies an unknown transform name. Verify the transform name in the accessory's parameter file.

- 0122 *Setname* does not have key data
- This message indicates that the specified set does not contain any key data. The DBEngine returns this result in response to a call to the DBKeyDataRemap entry point.
- 0123 Unable to link to *libraryname* *reason*
- DBEngine tried to link to the indicated library but was unable to do so because of the stated *reason*. If this is unexpected and persists, contact Technical Support.
- 0124 Missing format *formatname*
- The specified format does not exist. Create the format or enter the name of a different format.
- 0125 GetStatus error: *reason*
- This message describes errors returned from calls to the system GetStatus function.
- 0126 *filename* is not a file
- This message indicates that the specified filename is actually a directory name.
- 0127 Error reformatting *itemname* of *datasetname* with user value *uservalue*
- This message indicates a problem with the specified reformatting routine. Check the reformatting routine for errors.
- 0128 Reader-specified dataset *datasetname* failed VERIFY test for record *recordloc* in file *afn*
- This message occurs when a flat file record returned by a FileExtract Reader library fails the VERIFY test associated with that structure. Correct the FileExtract Reader library or the VERIFY test in the DASDL and then rerun FileExtract. Refer to the *Databridge FileExtract Administrator's Guide* for more information.
- 0130 Remote DBServer protocol level is *ServerLevel* but needs to be *RequiredLevel*
- This message indicates that the remote DBServer protocol level is lower than what is required. The remote system needs to run a newer version of DBServer.
- 0131 SetStatus error: *description command*
- This message reports an error message returned from the SETSTATUS system function and the attempted command.
- 0132 *virtualds* is derived from *basedataset*, so they must be selected together
- The accessory attempted to clone a virtual dataset *virtualds* without also cloning the dataset *basedataset* from which it is derived.
- 0133 Invalid DBSupport title: *s'uppporttitle'*
- The accessory specified an illegal title for the Support Library.
- 0134 No license for *platform product*
- The DBEngine parameter file does not contain a valid key for running the product on the platform.
- 0135 0135 License for *platform product* expired
- The DBEngine parameter file contains a key for the product but the license expired on the indicated date.
- 0136 License for *platform product* will expire *timestamp*

The DBEngine parameter file contains a key for the product but the license will expire soon on the indicated date. This is only a warning message.

0137 Invalid audit file prefix: *p'refix'*

The accessory called the DBAuditPrefix entry point with an illegal audit file prefix.

0138 Can't track *embeddeddataset* because parent records do not have a valid AA

The accessory selected an embedded dataset in fixup or normal mode but the parent dataset does not have record addresses usable for associating an embedded record with its parent record.

0139 Invalid file title: *f'ilettitle'*

DBEngine encountered an illegal file title for the DMSII DESCRIPTION file or another file.

0140 Invalid AFN: *afn*

The accessory supplied an illegal audit file number. For DMSII audit files, valid audit file numbers range from 1 to 9999.

## 0900-0999 (Accessories)

0901 Invalid filter name: *name*

The FILTER name specified in the parameter file is not the name of a filtering routine in the Support Library. Either correct the name in the parameter file or use the GenFormat program to define the filter to the Support Library and recompile it. Make sure that you entered the correct library name for the SUPPORT parameter.

0902 Invalid format name: *name*

The FORMAT name specified in the parameter file is not the name of a formatting routine in the Support Library. Either correct the name in the parameter file or use the GenFormat program to define the format to the Support Library and recompile it. Make sure that you entered the correct library name for the SUPPORT parameter.

0903 Support SETUP failed; Accessory version = *version*

The DBSETUP entry point in the Support Library returned an error. Typically, this means one of the following:

You entered an incorrect SUPPORT library title in the parameter file.

—or—

The Support Library needs to be recompiled.

0904 Interface version mismatch

DBEngine and the accessory were compiled with different versions of the Databridge API. Copy the correct version of the Databridge software and rerun the accessory. If you are running a user-written accessory, recompile it with the correct DBInterface file (*SYMBOL/DATABRIDGE/INTERFACE*).

0905 Another program opened *filename* EXCLUSIVE

*filename* is the name of a parameter file. This message appears when an accessory attempts to access its parameter file and some other program has opened it EXCLUSIVE. Usually this indicates that another copy of the accessory is already running and using the same parameter file. Determine which program has the parameter file opened EXCLUSIVE and either wait for it to finish or manually terminate it. Then, rerun the accessory.

0906 Syntax error in parameter file

This message indicates that there is an error in the parameter file. It is preceded by a more descriptive message indicating the line number and column number of the syntax error.

0907 Missing entry point *entrypoint* in DBSupport

The accessory tried to use an entry point in DBSupport that was not in the Support Library. Recompile DBSupport or correct the user-written accessory and try again.

0908 Duplicate entry for STR# *structurenumber*

This message indicates that there is a duplicate structure number (and record type) in the accessory's parameter file. This is usually the result of a mistake in editing the file. Correct the parameter file and rerun the accessory.

0909 Initialization failed

The accessory detected errors when it tried to initialize. See the accessory's report file for the specific error message.

0910 Duplicate option (at line *nnn*): option

Check the following:

Does the parameter file contain a duplicate entry for a SOURCE (for example, two SUPPORT entries)? If so, remove the duplicate entry.

Does the parameter file contain a duplicate entry for a STOP command (for example, two STOP BEFORE commands)?

0911 Parameter file not available: *filename*

The accessory was unable to find the parameter file or the file is not available to be opened exclusively. Copy the sample parameter file as the name of the missing file and rerun the accessory.

If DBTwin returns this error, copy the sample parameter file as DATA/TWIN/databasename/CONTROL, modify it so that it matches the DBServer SOURCE on the primary system, and rerun Databridge Twin.

0912 Invalid time limit=*hh:mm*

The time you entered for the STOP option is invalid for any of the following reasons:

- ◆ The value for hours (*hh*) is outside of the valid range of 0 to 23.
- ◆ The value for minutes (*mm*) is outside of the valid range of 0 to 59.

Correct the time and run the accessory again.

If DBTwin returns this error, see "STOP" in the *Databridge Twin Administrator's Guide*.

0913 Invalid date limit=*mm/dd/yyyy*



The date you entered for the `STOP` option is invalid for any of the following reasons:

- ◆ The day is invalid for the specified month (for example, February 30, September 31)
- ◆ The month is 0 or greater than 12
- ◆ The year is between 100 and 1970

Correct the date and run the accessory again.

0914 Invalid number of days = *days*

The number of days you entered for the `STOP` option (`DBTwin`) is invalid because it caused the date to be before 1970. Correct the number and run the accessory again.

0915 Time should be 14 digits: *time*

A data set entry in the parameter file contains a date/time that is not 14 digits long. Enter it using the following syntax: `YYYYMMDDhhmmss`

0916 Error writing to *filename*

This message indicates a problem with a `WRITE` command on one of the accessory's output files. This problem might be a hardware problem. If the hardware is functioning properly and the error message persists, contact Technical Support.

0917 *n* Errors in parameter file *filename*

The accessory detected *n* errors in the parameter file. Correct the errors and rerun the accessory.

0918 Line *n* Error

The parameter file contains an error on line *n*. Correct the error and rerun the accessory.

0919 Error reading *filetitle*

The accessory encountered an error while trying to read *filetitle*.

0921 EOF reading *filetitle*

The accessory encountered the end-of-file while trying to read *filetitle*.

0999 Fatal error

The program or accessory encountered an error condition that won't allow it to continue processing. The program will terminate. Some of these conditions include the following:

- ◆ Syntax error in the parameter file
- ◆ An error retrieving the creation date and time
- ◆ Software version mismatch
- ◆ An error during initialization
- ◆ A failure in actually switching the audit file

## 1000-1099 (DBServer)

1001 Unable to assume configuration `usercode:usercode`

DBServer could not use the usercode specified in the DBServer parameter file. Check both of the following:

Make sure that DBServer has the TASKING privilege. Otherwise, it cannot assume the usercode specified in the DBServer parameter file.

Make sure that the usercode specified in the Server parameter file has visibility to the DATA/ENGINE/CONTROL file.

1002 Unable to link to *SUPPORTlibrary (reason)*

DBServer could not attach to the Support library specified in the DBServer parameter file. Either the Support library name is misspelled in the DBServer parameter file, or the specified Support Library is missing.

1006 Invalid host password

This message indicates that DBServer has a password specified in its parameter file (KEY=password) and that a Databridge Client user has entered the wrong (or no) password.

1007 No default source defined

This message applies to Databridge 1.0 Clients. It indicates that no default data source is defined in the DBServer parameter file. To do this, add the keyword DEFAULT to the SOURCE entry in the DBServer parameter file.

1008 Data source name required

This message indicates that the DBServer parameter file does not contain a data source name. Make an entry for SOURCE in the parameter file.

1009 Data source *sourcename* not found

The data source that a Databridge Client user specified with the dbutility command does not exist. Check the spelling of the data source name. The data source name must match the entry for SOURCE in the DBServer parameter file.

1010 Structure *structurename* is not viewable

This message indicates that a Databridge Client tried to select one or more data sets, but they are filtered out by the Support Library. Use the Client DEFINE or REDEFINE command to reload the database layout.

1011 Interface version (Databridge *Engineversion*) mismatch

DBServer and DBEngine were compiled with different versions of DBInterface. Copy the correct software from the release tape.

1013 Invalid function requested: *functioncode*

This message indicates that the Client supplied an invalid function code in its request. If the error persists, contact Technical Support.

1014 I/O error on port *portnumber*: *description*

DBServer encountered an I/O error trying to use the network transport, for example, TCP. The description indicates the type of error. This error is often the result of the Databridge Client terminating abnormally or the operator terminating the network transport. After correcting the cause of the error, try rerunning the Databridge Client.

1015 Operator terminated Worker

An AX QUIT command was issued to a Worker that was receiving data from an audit file. The Worker will keep the connection open, but return this message and then stop at the end of the current transaction group.

1016 Invalid Server option number: *number*

This message indicates that the client supplied an invalid Server option. Make sure you have the proper version of the host Databridge software for the Client you are using.

1017 Assertion failed @ *linenumber*

DBServer detected an unexpected data value. If this condition persists, report it (including the line number) to Technical Support.

1018 Unoffered subport opened: *subport*

A program such as the Databridge Client opened a subport of DBServer's network transport file that DBServer had not offered for use. This unexpected situation could cause DBServer to run more Workers than specified in the parameter file.

1019 Invalid FILESTATE at open of *transport port [subport]; description*

A program such as the Databridge Client opened a subport of DBServer's network transport file that had a file state other than CLOSED, as indicated by description. This may indicate a network transport problem.

1020 Bad port number syntax

The port number configured in the DBServer parameter file is greater than 65535. Enter a correct port number and rerun DBServer.

1021 Subport *subport* was already open

DBServer initiated a duplicate Worker for the subport. If this error persists, contact Technical Support.

1022 Parameter file had *nnn* errors

The DBServer parameter file contains errors. Correct the errors and then rerun DBServer.

1023 Invalid command: *command*

The AX command you entered is not a valid command. Enter a valid command.

1024 Source not found (at line *nnn*): *sourcename*

The target source in the SOURCE...LIKE... command (at line number *nnn* in the DBServer parameter file) is invalid. Enter a valid target source.

1025 OPEN error on transport port [*subport*]: *description*

DBServer encountered an I/O error trying to open the subport of the network transport. The description indicates the type of error.

1026 DATABASE must be specified for SOURCE *sourcename*

The DBServer parameter file does not have a valid database specified in the SOURCE option.

1027 PORT or ENDPOINT must be specified

The DBServer parameter file did not contain a PORT or ENDPOINT declaration indicating the type of network transport to use. Fix the parameter file and rerun DBServer.

- 1028 At least one SOURCE must be specified  
You must specify at least one SOURCE entry in the DBServer parameter file.
- 1030 Invalid file handle: *number*  
Enterprise Server requested a file operation with an invalid handle number.
- 1031 Expected number: *text*  
The operator entered an AX command to DBServer that requires a number (for example 1234 AX BUFFER *nnn*), but instead entered text. Re-enter the AX command with the correct number.
- 1032 Default source already specified (at record *nnnnnn*): *DEFAULT*  
The DBServer parameter file contains more than one SOURCE entry containing the DEFAULT option. Remove the DEFAULT option from all SOURCE entries except one.
- 1033 Invalid client host *host* for source  
This message indicates that a Client tried to connect to a SOURCE from a host that was not listed in the HOST declaration.
- 1034 Row *rownum* of *filetitle* has I/O errors  
The indicated file row (area) is locked out with an I/O error.
- 1035 XDR error @ *linenumber*  
DBServer detected invalid (or missing) data in the message to/from the Databridge Client. Make sure you have compatible versions of the host and Client software. If this message persists, contact Technical Support and be sure to include the DBServer line number.
- 1036 Unable to send response @ *linenumber*  
When DBServer attempted to send a response message to the Databridge Client, it received an error. This could be due to an abnormal termination of the Client or a network problem.
- 1037 Read\_Callback interrupted by client input  
DBServer was in the middle of sending the updates in a transaction group when it received an input message from the Client. When this happens, DBServer discards the rest of the transaction group and tries to process the input message.
- 1038 Workers can not exceed *nn*  
The message indicates that the number of workers specified in the DBServer parameter file or the AX WORKERS command is set too high. The maximum number of workers you can set is 100.
- 1043 File not found: *filetitle*  
The client requested a file that was not found on the host system.

## 1100-1199 (DBEnterprise)

- 1100 I/O error on *filename*: *description*  
Enterprise Server encountered an I/O error on the indicated file.
- 1101 Open error on *filename*: *description*

Enterprise Server was unable to open the indicated file. This is not necessarily a fatal error. For example, there are no cache files present the first time Enterprise Server is run to cache a particular source.

1102 Error reading from *filename: description*

Enterprise Server encountered an error trying to read the indicated file.

1103 Error writing to *filename: description*

Enterprise Server encountered an error trying to write the indicated file.

1104 Allocation of *number* bytes failed

The operating system could not provide the requested number of bytes of memory from the heap.

1105 Deallocation of *number* bytes failed

The operating system could not return the specified number of bytes of memory to the heap.

1106 Selection limit *maxselects* exceeded: *numselections*

The Databridge Client tried to select more data sets than the maximum allowed.

1107 WinSock initialization failure. *description*

Windows Sockets couldn't be initialized. This means TCP/IP is unavailable

1108 Invalid command line parameter: '*parametername*'

Check the spelling of the parameter you are entering, and make sure that parameter is valid for the command.

1109 Client protocol level (*clientlevel*) must be at least *minimumlevel*

The Databridge Client's protocol level is obsolete and must be upgraded. To access Enterprise Server, the client's protocol level must be the equal to or greater than the specified *minimumlevel*.

1110 Filter routine failed

An error was encountered in the filtering routines. Remove the associated filter files from the Enterprise Server Config directory and rerun the client:

*sourcename(updatelevel).filter*

If the error persists, contact Micro Focus Technical Support.

1111 Trouble closing audit file *afn*

Enterprise Server encountered an error while trying to close a cache file.

1112 DBSelect requires data set name (*#structurenum*)

The data set selection Remote Procedure Call from the client did not include a data set name, which is required. *structurenum* is the DMSII structure number.

1113 Source does not have a base

Enterprise Server could not find the base source associated with a local source. Delete the local source, add the base source, and then create the local source from it.

1114 Audit file *afn* corrupted

A cached audit file is corrupted. Remove it and any later cached audit files for that source and then run Enterprise Server with the CACHE command to rebuild them.

1115 Corrupted data in data set block *blocknum*

A file block in an ORDERED data set is corrupted. Report this to your DMSII database administrator.

1116 Incorrect MCP label *headerfield* = 'value' on drive *drivenum* unit *unitnum*

The disk label is incorrect for an MCP disk. Enterprise Server will not try to use that drive for direct disk cloning.

1117 Structure #*structurenumber* format level mismatch: Filter (*filterlevel*) vs. record (*datasetlevel*)

The loaded filter has a different data set format level than the record currently being filtered. Remove the associated filter files from the Enterprise Server Config directory and rerun the client:

*sourcename(updatelevel).filter*

If the error persists, contact Micro Focus Technical Support.

1118 Source *sourcename* does not have a cache

You attempted to use the Enterprise Server cache command for a source that does not have caching set up yet. This error may appear in the log the first time the source is cached but Enterprise Server will continue creating the cache files.

1119 Audit file *afn* is not open

The cached audit file was expected to be open but was not. If this error persists, contact Micro Focus Technical Support.

1120 Unrecognized file type version in *cachefile*

The cache file was created by an unsupported version of Enterprise Server or the file is not a cache file at all. Remove the file(s) from the cache directory before reprocessing.

1121 Audit location not found: AFN *afn*, ABSN *absn*, Inx *index*

The client specified an audit location that was missing or incorrect. Either correct the client control tables or reclone the database.

1122 DBEnterprise can not clone an ALTERed data set: *datasetname*

Enterprise Server cannot clone altered data sets that require a REFORMAT routine. These data sets require special formatting routines on the mainframe, so they can be cloned only by the Databridge Host software (unless you create a user-written .dll to perform the reformatting).

If there is an altered data set that you don't need in one of your local (filtered) Enterprise Server sources, right-click it and select **Exclude all rows**. The red square appears in the data icon, indicating that it will be omitted from replication.

1123 Family *familyname* has no local disk units

Enterprise Server does not have visibility to the disk units comprising the indicated family name. It will not use direct disk cloning to access the data on that family.

1125 Checksum error reading data set

The block checksum in a data set was incorrect and therefore the data in the block is suspect. Notify your DMSII database administrator. Enterprise Server will not use direct disk cloning to process this data set.

1127 Unable to access configuration directory. Please use setup.exe to install DBEnterprise

The Config directory is created by Enterprise Server during a successful installation. Reinstall Enterprise Server to create the directory and the associated registry keys.

If the registry keys were incorrect and the directory is still on disk, the setup program will preserve the sources you have already defined.

1128 Unable to access configuration file '*filename*'

Make sure that the configuration text file you are trying to import is in the Enterprise Server Config directory.

1129 Errors in configuration file *filename*

Review the configuration log file to determine where the error occurred.

1130 Unable to determine client IP address

Enterprise Server was unable to retrieve the IP address associated with a specified host name. The operating system was unable to reach a DNS to convert the host name to an IP address. This might be due to firewall restrictions.

1131 Virtual data set *datasetname* is derived from *datasetname*, so they must be selected together

You must select both of the specified data sets. To include both of the data sets in a local source, right-click and select Include all rows for each data set. The green square appears in the data icon, indicating that it will be included in replication.

The Databridge Client must select both data sets during replication. For example, you cannot clone just the virtual data set or just the data set it is derived from.

1132 Audit file *afn* audit level *auditlevel* is not supported

The audit file that Enterprise Server reads from the MCP server has an unsupported audit level and therefore can't be interpreted properly. Enterprise Server will try to request the audit data from DBServer.

1133 Host '*hostname*' interrupted data transmission

The specified host transmitted a new request before Enterprise Server finished responding to the previous request. This caused Enterprise Server to discard the rest of the previous response. This error could indicate problems in the network configuration or hardware.

1135 Operator terminated *programname*

Someone stopped Enterprise Server. It terminated at the first quiet point after receiving the quit command. If you run the Databridge Client again it will begin at that quiet point

1136 Unable to load *filename* filter for update level *updatelevel*

The update level stored in the filter did not match the update level number in the filename. If this is a fatal error, delete *filename* and Enterprise Server will regenerate it the next time you run the Databridge Client.

1137 Unable to load *sourceinfoname* source information

Enterprise Server was unable to find the source information file for the requested source. The source information file should be in the Config folder and have the file extension ".source". Correct the source name if it is in error. Otherwise, use Enterprise Server to define it.

1138 Unable to start *description*

Enterprise Server was unable to start a process or thread. It will not be able to use direct disk cloning (or whatever the description indicates).

1139 Unsupported function

The requested RPC is not supported in the current release of Enterprise Server. This can occur in some situations with "cascading" Enterprise Server.

1140 Unable to change to directory *directoryname*

Either the directory is missing or Enterprise Server does not have access rights to change to it.

1141 XML parser error: *description*

The flat file schema file had syntax or semantic errors. Correct the schema file.

1143 Unable to find *filename* (*reason*)

A necessary file is missing. Restore the file or regenerate it.

1144 Fatal error: Unable to open a log file for *functionname*

Enterprise Server was unable to create a log file in the Logs subdirectory. Either the subdirectory is missing or corrupted or out of space. Enterprise Server will not continue if it can't write a log file.

1145 Unexpected EOF reading from *filename*

Enterprise Server encountered the end of the input file before it found data it expected to be there. If the filename is `LastSchema.txt`, make sure that it contains a valid number.

1146 Update level requested (*updatelevel*) > LastSchema update level (*currentupdatelevel*)

A flat file "audit" or clone subdirectory requires a schema level greater than the current schema level, which is specified in the `LastSchema.txt` file. Provide a schema for the desired update level and change the `LastSchema.txt` file to contain the new level.

1147 Update level is *schemaupdatelevel* not *clientupdatelevel*

The flat file schema update level is *schemaupdatelevel* but the `DATASOURCES` table in the client database had update level *clientupdatelevel*. You must provide the proper flat file schema or redefine and reclone the client database.

1148 SQL ODBC initialization failed

When Enterprise Server tried to process the `VALIDATE` command it was unable to access ODBC. Make sure that ODBC is supported in your Windows operating system.

1149 SQL Server login failed

The username/password was missing or incorrect on the `VALIDATE` command. Make sure that the username is valid for the Client database.

1150 Unable to connect to database '*databasename*'

Enterprise Server cannot connect to the Databridge Client database. Verify that the Client, server, and relational database are running.



- 1151 Unable to append SQL text '*text*'
- While constructing an SQL command, ODBC returned an error. If this error persists notify Micro Focus Technical Support.
- 1152 Unable to execute SQL command: *command*
- The SQL command failed probably due to a syntax error or security restrictions. If the command has a syntax error, notify Micro Focus Technical Support. Otherwise correct the problem and rerun the command.
- 1153 SQL Free Statement failed
- The SQL Free command failed. If this error persists, notify Micro Focus Technical Support.
- 1154 SQL Prepare Statement failed for: *statement*
- The SQL Prepare command failed. If this error persists, notify Micro Focus Technical Support.
- 1155 SQL Fetch failed for: *statement*
- The SQL Fetch command failed. If this error persists, notify Micro Focus Technical Support.
- 1156 SQL variable bind failed
- The dbbind function failed when trying to associate local variables to SQL columns. If this error persists, notify Micro Focus Technical Support.
- 1157 SQL data conversion failed
- Enterprise Server was unable to convert an SQL variable to a text string for comparison with a text file. If this error persists, notify Micro Focus Technical Support.
- 1158 Unexpected SQL data type: *datatype*
- The VALIDATE command encountered an SQL variable type that is not currently supported in Enterprise Server. If you need this data type, contact Micro Focus Technical Support with this error for a possible new feature enhancement.
- 1159 No data items found for '*datasetname*' in source '*sourcename*'
- This can occur when all of the data items were filtered out by DBServer (via the Databridge Support library), or that the local filter has filtered out the data items.
- 1160 Table '*tablename*' key '*keyvalue*' row not found for record number *recnum*
- The VALIDATE command read a record from the flat file but no matching row was found in the client database. Most likely the record was filtered out.
- 1161 Table '*tablename*' key '*keyvalue*' data mismatch for record number *recnum*, offset *offset*
- The VALIDATE command has found a mismatch between the record in the input text file and the row that was stored in the relational database. Report this to Micro Focus Technical Support.
- 1162 Table '*tablename*' key '*keyvalue*' duplicate rows
- The specified key was found in more than one row in the specified table. Check to see that the index in the Client database is UNIQUE. You will probably have to reclone the table after the problem is corrected.
- 1163 Unable to connect to server *servername* at port *portnumber*

This can occur in the following circumstances:

- ◆ The destination server is down.
- ◆ The server name or IP address was entered incorrectly
- ◆ Server access is prohibited by a firewall

1164 Winsock initialization failed

This is a Windows error indicating that there is a problem with TCP/IP.

1165 Error opening communication socket

Enterprise Server was unable to establish a TCP/IP connection with the remote host (either DBServer or a Databridge Client). Make sure the remote host is still responding to TCP/IP requests.

1166 Host '*hostname*' closed communication socket

Communication may have been interrupted on the specified host. This error occurs when the remote process terminates abnormally, such as a Ctrl-C on the Databridge Client or a DS on DBServer. Check the host, make sure it is running, and then try again.

When tracing is enabled in Enterprise Server, the log will contain these messages when the Client or DBServer disconnects. If replication is complete these messages can be ignored.

1167 Error sending network message

This message indicates a problem sending messages on the network. Replication will terminate immediately. Usually this means the remote process has terminated abnormally.

1168 Error reading network message

This message indicates a problem reading messages on the network. Replication will terminate immediately. Usually this means the remote process has terminated abnormally.

1169 Message sequence error (expected *requestnum* but received *responsenum*)

The response from the remote process has a sequence number different from the last request sent to it. This could indicate network errors or messages leftover from a previous replication run.

1170 Protocol level (*DBServerlevel*) must be at least *minlevel*

The DBServer protocol is too low to be compatible with Enterprise Server. A new release of the Databridge Host software must be installed.

1171 Unrecognized update type: *updatetypenum*

The DBServer has sent an update containing an unrecognized update type in the UpdateInfo. If this persists report it to Micro Focus Technical Support.

1172 Internal software error: *description*

Contact Micro Focus Technical Support.

1173 Invalid source type (*sourcetypenum*) for *sourcename*

The source information found in *sourcename*.*source* contains an unrecognized source type number. Source types can be Remote, File, or Local. The source information file has been corrupted. Remove it and use Enterprise Server to create it.

1174 Unable to determine client host name for *ipaddress*

The operating system is unable to retrieve the host name associate with an IP address. Typically this means the reverse-DNS is not accessible. Contact your network administrator.

1175 Please use `setup.exe` to install DBEnterprise

You are attempting to use an Enterprise Server feature that is not successfully installed.

1176 Tree item description removed

This is a status message informing you that the source you selected to remove was removed successfully.

1177 Audit Mirroring is not enabled for `sourcename`

Audit mirroring is disabled because the Mirroring parent directory option is blank.

1178 Audit read size (`length`) too short for block

The amount of data read from the audit file is less than the declared size of the audit block. The entire audit block has to be readable in order to process it. Enterprise Server will automatically retry the read after a brief waiting period. If this error persists, notify Micro Focus Technical Support.

1179 Audit block size (`length`) exceeds max (`maxlength`)

The value in the audit block size field exceeds the maximum size declared in the DASDL. Usually this error indicates that data read is not really an audit block, just leftover data on disk. Enterprise Server will automatically retry the read after a brief waiting period. If this error persists, notify Micro Focus Technical Support.

1180 Block first ABSN (`absn`) not = dup ABSN (`dupabsn`)

The ABSNs in the current audit block do not match. An audit block contains its ABSN in two places for integrity checking. Usually this indicates that the audit block has not been completely written yet. Enterprise Server will automatically retry the read after a brief waiting period. If this error persists, notify Micro Focus Technical Support.

1181 Username/password encryption error: `errmsg`

The Windows encryption routine encountered the error identified by `errmsg` when encrypting the username and password.

1182 Username/password decryption error: `errmsg`

The Windows decryption routine encountered the error identified by `errmsg` when decrypting the username and password.

1183 Logon to user '`username`' failed

The Windows logon using `username` and password failed. This error occurs when Enterprise Server attempts to use an invalid username/password pair. Correct the username and password in the appropriate configuration dialog box.

1184 Checksum error in `Afn afn Absn absn sector seg`

The computed checksum and the stored checksum for an audit block do not match. This error indicates this audit file has experienced some type of I/O error. Enterprise Server will automatically retry the read after a brief waiting period. If this error persists, notify Micro Focus Technical Support.

1185 Address Check error in `auditfile sector seg` found `address`

The audit block read at the specified sector has an AddressCheck value of address, when sector was expected. This error indicates this audit file has experienced some type of I/O error. Enterprise Server automatically retries the read after a brief waiting period. If this error persists, notify Micro Focus Technical Support.

1187 Invalid Julian date (YYYYDDD): *date*

The date conversion routines received a request to convert a Julian date but the supplied date was not a valid Julian date.

1188 Invalid Gregorian date (YYYYMMDD): *date*

The date conversion routines received a request to convert a Gregorian date but the supplied date was not a valid Gregorian date.

1189 Invalid type parameter '*param*'

The parameter value was not legal for the indicated parameter. Correct the parameter value and try again.

1190 DBServer protocol *level* does not support RPC *function*

A remote procedure call (RPC) was attempted for a function that is not supported by the attached DBServer. Upgrade DBServer (Databridge Host software) to a later release and try again.

## 1200-1299 (DBSpan)

1201 Found task *taskname* in audit

This informational message indicates that Span met the requirements set by the STOP option. It has found the task indicated in the STOP option and will terminate at the next quiet point.

1202 Exceeded audit file limit: *n*

This is an informational message that informs you that Span met the requirements set by the STOP option. Span has processed the number of audit files (*n*) indicated in the STOP option and will terminate at the next QPT.

1203 Exceeded audit timestamp limit: *time*

This is an informational message that informs you that Span met the requirements set by the STOP option. Span passed the timestamp indicated in the STOP option and will terminate at the next QPT.

1204 Encountered record limit: *n*

This is an informational message that informs you that Span met the requirement set by the STOP option. Span received the number of record updates indicated (*n*) in the STOP option and will terminate at the next QPT.

1205 Error repositioning *filename*

This message indicates a problem with a SEEK command on one of the Span output files. This problem might be a hardware problem. If the hardware is functioning properly and the error message persists, contact Technical Support.

1206 No dataset entries in parameter file

This message indicates that you did not select any data sets to be processed. Remove the percentage sign (%) in front of the status information for the data sets you want to replicate.

1207 `Creating new parameter file filename`

This message indicates that Span is creating the specified parameter file.

1208 `No header in existing file`

The Span parameter file indicates that the output data files will contain a header record showing the starting and ending audit locations contained in the file. Span tried to read an existing output file's header record but did not find it. It is likely that the `HEADER` option was changed to `TRUE` in the parameter file after the output file was created. If this is the case, transfer the output file to the target system and remove it from the source system so that Span does not find it. Then rerun Span.

1209 `Ending location too short in existing file`

The Span parameter file indicates that the output data files will contain a header record showing the starting and ending audit locations contained in the file. Span tried to read an existing output file's header record, but the area for the ending audit location information was too short.

Although a corrupted output file could cause this problem, it is more likely that the `HEADER` option was changed to `TRUE` in the parameter file after the output file was created. If this is the case, transfer the output file to the target system and remove it from the source system so that Span does not find it. Then rerun Span.

1210 `Output file audit location mismatch`

This message can occur when the following conditions are true:

- ◆ `HEADER = TRUE` in the Span parameter file
- ◆ You are running Span version 2.0 or later
- ◆ A Span `UPDATES` file is present

In this case, Span tried to append data to an existing output data file with `HEADER=TRUE` but found that the existing file's header doesn't match the current audit location. Span renames the file `.../BAD` and opens a new data file.

To avoid this message, move any `UPDATES` files created by a previous version of Span before you run the current version. If the current version of Span created the existing `UPDATES` file, then either the Span parameter file was replaced with an older version, or the `UPDATES` file was replaced with an older version.

1211 `Previous end location = auditlocation`

This is an informational message that tells you the audit file location (`AFN`, `ABSN`, `SEG`, and `INX`) according to the header record in an existing output file from which Span will start to gather updates. This message is displayed when there is a conflict between the audit location in the parameter file and the audit location in the header record of an output file.

1212 `Current start location = auditlocation`

This is an informational message that tells you the audit file location (`AFN`, `ABSN`, `SEG`, and `INX`) according to the header record in an existing output file. The header record corresponds to the first data record in the output file. This message is displayed when there is a conflict between the audit location in the parameter file and the audit location in the header record of an output file.

1218 `No starting location in existing file`

The Span parameter file indicates that the output data files will contain a header record showing the starting and ending audit locations contained in the file. Span tried to read an existing output file's header record but did not find the delimiter indicating the starting location. The output file is probably corrupted. Either remove it or replace it from a backup copy. Then rerun Span.

1219 No ending location in existing file

The Span parameter file indicates that the output data files will contain a header record showing the starting and ending audit locations contained in the file. Span tried to read an existing output file's header record but did not find the delimiter indicating the ending location. Most likely, the output file is corrupted. Either remove it or replace it from a backup copy.

1220 *parameter* must be greater than 0. Adjusting to *nnn*

The Span parameter file contained a negative value for *parameter*. Span automatically adjusts it to *nnn*.

1221 MAXRECSIZE not file-equated. Using default *nnn* bytes

This is a warning message. The Span parameter file says to write a consolidated output file to tape, but the WFL did not specify a MAXRECSIZE for OUTFILE. Tape files must have a fixed record size. Span will use a MAXRECSIZE of *nnn* bytes for OUTFILE. Records longer than that will be truncated.

1222 *datasetname* formatted record size *nnn* truncated to *mmm* bytes

This is a warning message. The output file record size specified for *datasetname* in the Span parameter file, *mmm*, is smaller than the size of a formatted record, *nnn*, for *datasetname*. The records will be truncated.

1223 *datasetname* attribute cannot exceed *nnn*. Adjusting to *mmm*

The output file record size, block size, or area size for *datasetname* (or DEFAULT) in the Span parameter file is larger than the mainframe size limit of *nnn* and will be changed to *mmm*.

## 1300-1399 (DBSnapshot)

1304 Invalid number of days=*days*

The number of days you entered for the STOP option is invalid because it caused the date to be before 1970. Correct the number and run Snapshot again.

## 1400-1499 (DBSupport)

1401 Format level mismatch for Str# *nnn* (Support=*supportlevel* Audit record =*formatlevel*)

A DBFORMAT routine in the Support Library was compiled with a different DESCRIPTION file than the one corresponding to the current audit file. Snapshot, Span, DBServer, and Databridge Twin will attempt to automatically recompile the support library when this error occurs. If that fails, recompile Support manually.

1402 Format cannot check for *null* or invalid data

The accessory set one of the data validation options but the format does not support data validation.

1403 DBSupport has update timestamp *supportupdatets* but database is *dbupdatets*

The Support library was compiled against a different database update level and needs to be recompiled.

1404 DBSupport has LINKS option *supporttruefalse* but current Databridge Engine parameter file has *enginetruefalse*

The Support Library was compiled with a different setting for the LINKS option than the current setting in the DBEngine parameter files. Either recompile Support or change the LINKS option in the DBEngine parameter files.

1405 DBSupport database *supportdbname [supporttdbts]* doesn't match *enginedbname [enginedbts]*

The Support Library was compiled against a different database than the one that DBEngine has open. Recompile the Support Library.

## 1500-1599 (DBAuditTimer)

1501 CREATIONTIME not returned

AuditTimer tried to get the creation time for an audit file but encountered this error. Notify Technical Support if this error persists.

1502 CREATIONDATE not returned

AuditTimer tried to get the creation date for an audit file but encountered this error. Notify Technical Support if this error persists.

## 1600-1699 (DBGenFormat)

1603 Duplicate name: at line *nnn*

You have used the same name twice (for example, two filters have the same name or a filter and a filter have the same name). All declared names in the GenFormat parameter file must be unique. Choose a new name for the duplicated name and rerun the GenFormat program.

1604 Translation not defined: *translationname* at line *nnn*

The translation table name you entered for a particular FORMAT declaration has not been defined. Check your spelling of the translation table name or create the translation table in the GenFormat parameter file.

1605 Duplicate definition: *optionname* at line *nnn*

This message indicates that you have entered the same option twice in a FORMAT declaration. *optionname* can be one of the following:

- ◆ PREFIX
- ◆ POSTFIX
- ◆ POSITIVE
- ◆ NEGATIVE
- ◆ UNSIGNED
- ◆ TEXT
- ◆ BOOLEAN
- ◆ SEPARATOR
- ◆ DECIMALPT
- ◆ PADDING
- ◆ OVERFLOW
- ◆ NULL
- ◆ TRUE
- ◆ FALSE
- ◆ FLOAT

In a FORMAT declaration, you can specify each option only once. Remove the duplicate option and rerun the GenFormat program.

1606 Variable format type *recordtype* already specified at line *nnn*

The specified variable format type *recordtype* has already been specified. This can occur when you declare a SELECT statement for a variable format data set and specify the same variable-format part twice. Combine or remove the duplicate clauses for the specified *recordtype* and rerun the GenFormat program.

1607 Duplicate SELECT for *datasetname* at line *nnn*

Within a FILTER, you have declared multiple SELECT statements referring to the same data set. Each data set can have only one SELECT statement in a FILTER. Combine the duplicate SELECT statements into a single SELECT statement and rerun the GenFormat program.

1608 Data item *dataitemname* not found at line *nnn*

This message can occur anytime you reference a data item, as in the following examples:

- ◆ When you declare an ALTER data set
- ◆ When you declare PRIMARY keys
- ◆ When you use a SELECT statement in a filter

In all cases, the data items you use in the parameter must have the same name they have in the DASDL or in a preceding ALTER or VIRTUAL data set. Correct the spelling of the data item and rerun the GenFormat program.

1609 *count* errors detected

This message indicates the number of errors the GenFormat program encountered in its parameter file.

1611 OCCURS *nnn* exceeds the maximum *mmm* at line *rrr*



The ALTER or VIRTUAL declaration contains a data item having OCCURS *nnn* but the limit is *mmm*. Change the OCCURS to a legal value.

1612 Version *version* compiled *compiletime*

*version* indicates the release of the GenFormat program and *compiletime* indicates when the GenFormat program was compiled. This is an informational message.

1613 Warning: Boolean strings *truestring* and *falsestring* are different lengths in *formatname*

The true and false strings must be the same length. This is a warning that the data items following a Boolean will be in different positions depending on whether the data item is true or false.

1614 Invalid for non-tailored. Restart compile with database name at line *nnn*

This message indicates that you referenced a data item in the parameter file. Any time you specify a data set or data item name in a FILTER, ALTER, PRIMARY KEY, etc., declaration, you must create a tailored support library.

1615 Name is too long: *name* at line *nnn*

The data set or data item name you have entered for an option exceeds the length allowed by DMSII. Currently, data set names are limited to 17 characters and data items names are limited to 30 characters. Correct the spelling of the name and rerun the GenFormat program.

**NOTE:** There is no limit on the size of user-declared names for FORMATS, FILTERS, TRANSFORMS, etc.

1616 Duplicate KEY for *datasetname* at line *nnn*

The parameter file contains two PRIMARY key declarations for the same data set. Each data set can have only one PRIMARY key declaration. Remove the duplicate PRIMARY key declaration and rerun the GenFormat program.

**NOTE:** A single PRIMARY key declaration can specify a compound key.

1617 Duplicate ALTER: *datasetname* at line *nnn*

You have already declared an ALTER statement for the specified data set name. You can use only one ALTER declaration per data set.

1618 Can't ALTER item in occurring group: *groupname* at line *nnn*

You cannot convert a data item that is part of an occurring GROUP. Instead, you must convert the entire GROUP. Refer to the *Databridge Programmer's Reference* for ALTER restrictions.

1619 Dataset *datasetname* not found at line *nnn*

This message can occur anytime you reference a data set, as in the following examples:

- ◆ When you declare an ALTER data set
- ◆ When you declare PRIMARY keys
- ◆ When you use a SELECT statement in a filter

Correct the spelling of the data set and rerun the GenFormat program.

1620 Record type *recordtype* not found at line *nnn*

The parameter file contains a SELECT statement that references a record type of a data set. The specified record type was not found in the DMSII DESCRIPTION file. Enter a valid record type.

**NOTE:** VIRTUAL data sets do not allow variable format records, and variable format records are not referenced in ALTER declarations.

1621 ALTERed data item *dataitemname* not allowed in WHERE clause at line *nnn*

You are attempting to use a converted data item in a WHERE clause of a SELECT statement in a filter declaration. Refer to the *Databridge Programmer's Reference* for ALTER restrictions.

1622 Duplicate structure number *number* at line *nnn*

You have used the same structure number for two VIRTUAL declarations. All virtual data sets must have a unique structure number. Also, the structure number for a virtual data set must not match the structure number of an existing, real data set or set. Choose a different structure number and then rerun the GenFormat program.

1623 Structure number must be > previous structure numbers: *number* at line *nnn*

This message indicates that you have tried to define a virtual data set with a structure number that was not greater than the structure numbers of existing, real data sets and sets and previously defined virtual data sets. Choose a larger structure number.

Guidelines for selecting structure numbers include the following:

- ◆ Do not use actual DMSII structure numbers.
- ◆ Once you declare a virtual data set, subsequent data set numbers must be greater than previously declared virtual data sets.
- ◆ Each structure number must be unique.

Complete details on virtual data set structure numbers are listed in the *Databridge Programmer's Reference*.

1624 Duplicate REFORMAT declaration at line *nnn*

This message indicates that there are duplicate reformat declarations in the GenFormat parameter file. Refer to the *Databridge Programmer's Reference* for REFORMAT restrictions.

1625 *dataitem* original size: *nnn* but REDEFINE size: *mmm* at line *nnn*

This message indicates that the REDEFINE size differs from the original size. The REDEFINE size must be the same as the original size. Refer to the *Databridge Programmer's Reference* for REDEFINE restrictions.

1626 REDEFINE of *dataitem* requires byte-alignment at line *nnn*

This message indicates that the original data item was not on a byte boundary but the REDEFINE data type requires it to start on a byte boundary (as for GROUP items). Refer to the *Databridge Programmer's Reference* for REDEFINE restrictions.

1627 *Dataitem* in WHERE clause must not have an OCCURS clause: *dataitem* at line *nnn*

This message indicates that the specified data item in the WHERE clause has an OCCURS clause, which is not allowed. Refer to the *Databridge Programmer's Reference* for WHERE clause restrictions.

1628 *subset* is not a (sub)set of *dataset* at line *nnn*

The USING clause of a SELECT statement refers to a set or subset (subset) that is for another data set. The USING clause must specify a set or subset of data set *dataset*.

1629 non-tailored Support, DBInterface level *n*  
Support update level *mmm*, DBInterface level *n*

These are informational messages. The first form indicates you are compiling the nontailored (generic) DBSupport. The second form indicates you are compiling a tailored DBSupport using database update level *mmm*. In both cases, *n* indicates the Databridge Interface level number, that is, the value of DBV\_VERSION of SYMBOL/DATABRIDGE/INTERFACE.

1635 Merged data item *dataname* is in a different parent group

To merge data items, all of the data items must be in the same parent group.

1636 Merged data item *dataname* is not adjacent to previous item

The specified data item does not start where the previous data item ends. To merge data items, all of the data items must be adjacent.

1637 VIRTUAL dataset *datasetname* must have DERIVED FROM *datasetlist*

The DERIVED FROM clause is missing from the virtual data set declaration in the GenFormat parameter file. (In previous releases, the DERIVED FROM clause was optional.) Add the DERIVED FROM clause to the virtual data set declaration and then recompile the tailored support library.

1638 Filter *filtername* excludes *datasetname*, from which *virtualdatasetname* is derived

A filter includes a virtual dataset but excludes its DERIVED FROM dataset clause. The filter must specify the DERIVED FROM clause.

## 1900-1999 (DBTwin)

1900 Please specify a SOURCE rather than CFPRIMARY

This message indicates that you are using an older version of the DBTwin parameter file. Use the current version of the DBTwin parameter file. Refer to the *Databridge Twin Administrator's Guide*.

1901 Option name ignored. Use SOURCE

This message indicates that you are using an older version of the DBTwin parameter file. Use the current version of the DBTwin parameter file. Refer to the *Databridge Twin Administrator's Guide*.

1902 Remote SOURCE cannot have 'option'

DBTwin is using a DBServer source that has a source option DBTwin does not support. Remove the option from the DBServer SOURCE declaration.

1903 Remote SOURCE must have 'option'

DBTwin is using a DBServer source that does not have an option that DBTwin requires. Add the option to the DBServer SOURCE declaration.

## 4000-4099 (DBDMSIIClient)

The DMSII Client can generate the following errors.

4002 Unable to connect to port *portnumber* at *hostname*: *errormessage*

The DMSII Client was unable to connect to the specified port. The error message text will explain why the error occurred.

4003 Unable to read port *portnumber* at *hostname*: *errormessage*

The DMSII Client was unable to read the specified port. The error message text will explain why the error occurred.

4004 Unable to write port *portnumber* at *hostname*: *errormessage*

The DMSII Client was unable to write to the specified port. The error message text will explain why the error occurred.

4005 DASDL database compile failed

The DMSII Client was unable to compile the DASDL file because it contains a syntax error. Correct the syntax error in `PATCH/DATABRIDGE/DMSIICLIENT/clientdbname/DASDL` and then rerun the DMSII Client. Refer to the *Databridge DMSII Client Administrator's Guide* for information about configuring the DMSII Client DASDL settings.

4006 Set generation failed

The generation of a SET failed during the clone of the dataset.

4007 DBServer protocol level (*x*) must be at least *y*

The DMSII Client requires a certain level of RPC protocol indicated by *y*. However, Databridge Server supports RPC protocol levels up to *x*. Install a newer version of Databridge Server that supports RPC protocol *y* or above.

4008 ClientLib compile failed

There is an error in compiling the DMSII Client library file. Do not edit `SYMBOL/DATABRIDGE/DMSIICLIENT/LIB`. Reinstall the DMSII Client software. If the file was not edited and it still gets a syntax error, contact Technical Support.

4009 Unable to link to *library\_name*: *reason*

The DMSII Client was unable to connect to the DMSII Client library. The error message text will explain why the error occurred.

4010 DMS Error: *errormessage*

A DMSII operation failed. Refer to your Unisys DMSII documentation to resolve this error.

4011 DMUtility dataset purge failed

DMUtility was unable to initialize the data set. Refer to your Unisys DMSII documentation to resolve this error. Verify that `SYSTEM/DMUTILITY` is available.

4012 Update errors: *nnn*

This is a summary message indicating *nnn* errors occurred while the DMSII Client was updating the client database. Check the previously displayed messages. Refer to your DMSII documentation to resolve the error.

- 4016 `SYSTEMSTATUS error errornumber`  
Contact Technical Support.
- 4017 `DBServer closed port unexpectedly`  
Databridge Server closed the network communications port unexpectedly. This could be due to the operator DSing the Worker or a program fault.
- 4018 `XDR error at line line#`  
The DMSII Client detected incorrect data or not enough data in the message it received from Databridge Server or in the message it is trying to send to Databridge Server. Contact Technical Support if this condition persists.
- 4019 `Missing filename`  
The DMSII Client skeleton DASDL file is missing. Refer to the *Databridge DMSII Client Administrator's Guide*.
- 4020 `Invalid AX command: invalid_command`  
There was a syntax error in the AX command you entered. Refer to the *Databridge DMSII Client Administrator's Guide* for valid AX commands.
- 4021 `dataset_name not found for DISABLE`  
An unknown data set was specified in the DISABLE or DISABLE ALL command in the DMSII Client parameter file. Correct the data set name and then rerun the DMSII Client. Refer to the *Databridge DMSII Client Administrator's Guide*.
- 4022 `dataset_name is disabled`  
This error occurs when you include one of the following commands as the second parameter of WFL/DATABRIDGE/DMSIIClient and the DMSII Client parameter file says to disable one or more of the data sets that are in [datasetlist]:
- ◆ `UPDATE [datasetlist]`
  - ◆ `CLONE [datasetlist]`
  - ◆ `STATUS [datasetlist]`
- For example, if DATA/DMSIIClient/CLIENTDB/CONTROL has: DISABLE BANK, BRANCH, TELLER and then you do: `START WFL/DATABRIDGE/DMSIIClient ("CLIENTDB", "UPDATE HISTORY, BRANCH")` you would get error 4022 because BRANCH is disabled.
- 4023 `Expected keyword not token`  
There is a syntax error in the second parameter of the WFL. Start the DMSII Client WFL with the proper syntax. Refer to the *Databridge DMSII Client Administrator's Guide*.
- 4024 `Unknown data set: dataset_name`  
A data set specified in the second parameter of the WFL does not exist in the client database. Start the DMSII Client with valid data sets in the second parameter of the WFL. Refer to the *Databridge DMSII Client Administrator's Guide*.
- 4025 `Invalid port number: portnumber`  
The PORT command in the DMSII Client parameter file contains an invalid port number. Correct the syntax error and then rerun the DMSII Client. Refer to the *Databridge DMSII Client Administrator's Guide*.

4026 Unrecognized command: *errormessage*

There is a syntax error in the optional second parameter in the WFL command. Rerun the WFL using a valid second parameter. Refer to the *Databridge DMSII Client Administrator's Guide*.

4027 Missing insertion point (@) in *dasdl\_patch\_file\_name*

The at sign (@) was removed from the DMSII Client DASDL file. Re-edit the DASDL file, and then rerun the DMSII Client. Refer to the *Databridge DMSII Client Administrator's Guide*.

4028 Duplicate insertion point (@) in *dasdl\_patch\_file\_name* lines *firstlinenbr* and *secondlinenbr*

There is more than one at sign (@) in the DMSII Client DASDL file. Remove one of the at signs and then rerun the DMSII Client. Refer to the *Databridge DMSII Client Administrator's Guide*.

4029 Use WFL/DATABRIDGE/DMSIICLIENT/COMPILEDB to compile *clientdbname*

The DMSII Client has generated the new DASDL for the Client database but the parameter file has the COMPILE = MANUAL option. Start the indicated WFL using a usercode having access to the compilers used in the WFL.

4030 Use WFL/DATABRIDGE/DMSIICLIENT/REORGDB to reorganize *clientdbname*

The DMSII Client has generated the new DASDL for the Client database but the parameter file has the COMPILE = MANUAL option. Start the indicated WFL using a usercode having access to the compilers used in the WFL.

## Status Messages

The following status messages may appear on the ODT and on the station that is running the accessory or component. Note that status messages are not processed by error handler routines.

Fatal error

The accessory encountered an error condition that won't allow it to continue processing. The program will terminate. Possible errors include:

- ◆ Syntax error in the parameter file
- ◆ Software version mismatch
- ◆ An error during initialization

Syntax error on line *linenumber* column *columnnumber*: '*token*' unexpected

The accessory detected a syntax error in the parameter file at the indicated line number and column number. Instead of the expected keyword or value, the accessory found *token*. Correct the parameter file and rerun the accessory.

Syntax error on line *linenumber* column *columnnumber*: Missing end quote mark

The accessory detected a syntax error in the parameter file at the indicated line number and column number. It expected to find the closing quotation mark (") for a string. Strings must start and close on the same line of the parameter file. Correct the parameter file and rerun the accessory.

Unexpected end of parameter file

The accessory encountered the end of the parameter file before finding required keywords or values. Add the necessary syntax to the parameter file and rerun the accessory.

Version *version* compiled *compiletime*

*version* indicates the release of the accessory and *compiletime* indicates when it was compiled. This is an informational message.

Workers can not exceed *maxnumber* (*Engine*)

This message indicates that the number of Workers specified in the DBEngine parameter file or in an AX WORKERS command exceeds the maximum allowable number.

Workers limit: *nn* (*Engine*)

This message indicates the Workers limit.

Workers unrecognized limit: *text* (*Engine*)

This message indicates that the Workers limit is incorrect. Additional text appears with this message, which explains the problem.

Snapshot: After modifying the parameter file, please rerun

When you run Snapshot for a database for the first time, it creates a parameter file. Use CANDE to edit this file so that the options have the proper values. After saving the file, rerun Snapshot.

Snapshot: Creating *filename*

You ran Snapshot without a parameter file so it is generating one called *filename* for future use. This run of Snapshot does not generate any data files.

Snapshot: *nnn* records extracted (*nnn* filtered out)

This is the response to an AX STATUS command for Snapshot. It shows how many records Snapshot has read so far and how many it has skipped because the filter procedure discarded them.

Unrecognized command: *text*

This message indicates that you entered an unrecognized AX command.

DBSpan: After reviewing the parameter file, please rerun

When you run Databridge Span for a database for the first time, it creates a parameter file. Use CANDE to edit this file so that the options have the proper values. After saving the file, rerun Databridge Span.

DBSpan: Audit location: AFN = *afn* ABSN = *absn*

This is the second line of the response to AX STATUS command. It indicates the audit file number and the audit block serial number currently being processed.

DBSpan: Audit time: *timestamp*

This is the third line of the response to an AX STATUS command. It indicates the original time of the transaction that Databridge Span last processed.

DBSpan: Databridge SPAN Initializing *time* on *hostname*

This is an informational message that tells you the time Databridge Span is starting and the name of the host it is running on.

DBSpan: Databridge SPAN Terminating *time*

This is an informational message that tells you the time Databridge Span is stopping. Databridge Span could be stopping because of an AX QUIT command or because the requirements have been met for the STOP option.

DBSpan: Existing output file has different attributes

Databridge Span has found an existing output file to append to but it has different file attributes (for example, BLOCKSIZE, AREASIZE, etc.) than what it computed. The existing file will be renamed with the .../OLDATTS node. Process this file on the target system. Then change the target system to expect the output file with the new file attributes.

DBSpan: Final audit location: time, AFN=*nnn*, ABSN=*nnn*

This is an informational message that tells you the point in the audit file at which Databridge Span stopped processing. This information is also located in the NOTE attribute of each Databridge Span output file.

DBSpan: *nnn* records replicated (*nnn* skipped)

This is the first line of the response to an AX STATUS command. It indicates the number of records written to the output files and the number of records that were skipped because the filter procedure discarded them.

DBSpan: Reading parameter file *filename*

This is an informational message that gives you the name of the Databridge Span parameter file.

DBSpan: Renaming to .../*name*

Databridge Span has encountered a problem file and is renaming it as follows:

```
Appending /OLDATTS
Appending /BAD
/OLDATTS
```

Databridge Span appends /OLDATTS to indicate that the existing file has different record sizes than the data that the accessory is attempting to write. This can occur after a reorganization. In this case, do the following:

- ◆ Process the old files on the target system.
- ◆ Remove them on the source system.
- ◆ Prepare the target system to receive data files that have the new layout.

```
/BAD
```

Databridge Span appends /BAD to indicate that the existing file ending audit location does not match the current starting location. If the files were created before changing the HEADER option to TRUE in the Databridge Span parameter file, follow the same procedure as above, but in place of step 3, prepare the target system to receive data files containing the header.

DBSpan: Terminating at next quiet point

An operator has given Databridge Span an AX QUIT. Databridge Span will terminate at the next quiet point.

DBSpan: Thruput: *nnn* records/sec. CPU

This is an informational message that tells you the number of data records that Databridge Span processed per second of CPU time.

DBSpan: Warning: *datasetname* block size *nnn* exceeds limit of *mmm*



The output file block you specified in the Databridge Span parameter file for *datasetname* is larger than the system block size limit of *mmm*.

DBSpan: Warning: *datasetname* block size adjusted to *nnn*

Span Accessory revised the output file block size to *nnn* to make the block size within the mainframe limit.

DBSpan: Warning: previous *limittype* limit ignored

The parameter file contains duplicate settings for the STOP option. *limittype* can be either Time or Task name. Specify only one task name and one time limit. If you specify a second task name or a second time limit, Databridge Span discards the first one and uses only the second one.

DBSpan: Unrecognized command: *text*

This message indicates that you entered an unrecognized AX command.

## Windows Event DBDirector Messages

The following messages can appear in the Windows Application Event log associated with source *DBDirectorrelease* (for example, *DBDirector66*) when running Enterprise Server. Message numbers 30005-30013, 30112, 30117, and 30125 in the following list are for information only *and do not indicate an error*. Some of the messages have a suffix of the form

*winerror winerrormsg*

where *winerror* is a Windows-defined error number and *winerrormsg* is the associated error text returned by the Windows *FormatMessage* system call.

30001 DBDS_Unauthorized	Administrator rights are required for controlling DBDirector.
30002 DBDS_BadParam	Bad DBDirector parameter
30003 DBDS_NotInstalled	DBDirector not registered
30004 DBDS_StatusFailed	Unable to retrieve DBDirector status
30005 DBDS_ContinuePending	DBDirector continue pending
30006 DBDS_PausePending	DBDirector pause pending
30007 DBDS_Paused	DBDirector paused
30008 DBDS_StartPending	DBDirector start pending
30009 DBDS_StopPending	DBDirector stop pending
30010 DBDS_Installed	DBDirector registered
30011 DBDS_Removed	DBDirector unregistered
30012 DBDS_Started	DBDirector started
30013 DBDS_Stopped	DBDirector stopped
30014 DBDS_InstallFailed	DBDirector register failed
30015 DBDS_RemoveFailed	DBDirector unregister failed

30016 DBDS_StartFailed	DBDirector start failed
30017 DBDS_StopFailed	DBDirector stop failed
30018 DBDS_AlreadyStarted	DBDirector already started
30019 DBDS_AlreadyStopped	DBDirector already stopped
30020 DBDS_StillActive	DBDirector query still active
30021 DBDS_Unknown	DBDirector unknown result <i>num</i>
30022 DBDS_NoDBDirector	DBDirector program not found
30100 DBDS_EventLogOpenErr	Unable to open EventLog key
30101 DBDS_EventLogKeyErr	Unable to create EventLog subkey
30102 DBDS_EventFileErr	Unable to create EventMessageFile value
30103 DBDS_EventTypesSuppErr	Unable to create TypesSupported value
30104 DBDS_NoModuleName	Unable to get module name for install
30105 DBDS_ServiceMgrErr	Open Service Control Manager failed
30106 DBDS_ServiceOpenErr	Open Service failed
30110 DBDS_ServiceCtrlErr	Start Service Control Dispatcher failed
30111 DBDS_BadCommandOp	Unrecognized command line option <i>option</i>
30112 DBDS_ServiceCtrl	Calling Service Control Dispatcher. Please wait.
30113 DBDS_WinSockErr	[WinSock error <i>num</i> ] <i>errmsg</i>
30114 DBDS_NoListenPort	Databridge registry entry ListenPort not found. Using default port <i>num</i> .
30115 DBDS_NoProgFiles	Databridge registry entry ProgFiles not found
30116 DBDS_NoConfigFiles	Databridge registry entry ConfigFiles not found
30117 DBDS_EventRunServer	DBEnterprise for client at host ' <i>hostname</i> ' Process: <i>id</i> , thread: <i>num</i> , socket: <i>hexnum</i> ConfigFiles: <i>dir</i> , Command: <i>command</i>
30118 DBDS_RunErr	CreateProcess failed for <i>processname</i>
30121 DBDS_BlockingReset	Reset to blocking mode failed
30122 DBDS_SocketOpenErr	Can't open TCP socket
30123 DBDS_SocketBindErr	Can't bind local address to socket
30124 DBDS_ListenErr	Listen failure
30125 DBDS_EventListen	Listening for Databridge client connections on port <i>num</i>
30126 DBDS_MgrReportErr	Service manager error
30127 DBDS_WaitErr	Event wait failure
30129 DBDS_WinSockInit	WinSock startup
30130 DBDS_WinSockTerm	WinSock cleanup

# 2 Databridge Client Messages

This section lists errors and messages that may occur when running the Databridge Client. Messages generally appear onscreen and in the log file. If tracing is enabled, messages are also captured in the trace file. (The verbose tracing option TR\_VERBOSE generates a large number of low-level messages which only appear in the trace file.)

Messages are listed here in alphabetic order. To search for a message that starts with the name of some element of your system (such as a file or host name), use other words in the message as search criteria. Actual messages contain names instead of variables; therefore, your message may not appear in the order you expect. Additionally, messages that appear in more than one section of this guide will be found under "section\_name". Messages that are common to most commands are listed in a separate section labeled *Common Messages*.

---

**NOTE:** Information written to the log file at the start of a Client run is very useful for troubleshooting. Therefore, any time you contact Micro Focus for Technical Support, always include the actual, complete log file (versus a screen capture) for a Client run. Screen captures do not contain most of the necessary information required to troubleshoot problems.

---

For a list of trace messages and exit codes, see the appendices of *Databridge Client Administrator's Guide*.

## In this Chapter

- ♦ [“Command Messages” on page 51](#)
- ♦ [“Client Errors” on page 96](#)
- ♦ [“Usage Errors” on page 149](#)
- ♦ [“Host Messages That Appear on the Client” on page 150](#)
- ♦ [“Sockets Errors” on page 155](#)
- ♦ [“ODBC Errors” on page 156](#)
- ♦ [“OCI Errors” on page 157](#)
- ♦ [“Client Warnings” on page 158](#)
- ♦ [“Messages Associated with Client Exit Codes” on page 186](#)

## Command Messages

This section includes messages that occur when you issue a command using either the command-line client or the Client Console in conjunction with the Client Manager Service.

## Common Log Messages

These messages, common to almost every command, are listed in the section below instead of repeating them for each command. We also include a few common messages that are generated at the start of the run for the commands that connect to the Databridge Server (DBEngine or Enterprise Server). These commands include `define`, `redefine`, `process`, `clone`, `switchaudit` and `tcptest` commands. These messages are marked as "(Server connection only)".

All garbage data successfully flushed

(Server connection only) This message, which appears in the trace file when the TR\_VERBOSE bit in the trace mask is set, indicates that the Client was able to read all of the false TCP data after receiving an incorrect length response to the test pattern. It always follows the message "Flushing garbage data". This message does not appear under normal circumstances.

ATM\_ECHO: Pattern = '*string1*', Response = '*string2*'

(Server connection only) This message, which appears in the trace file when the TR\_VERBOSE bit in the trace mask is set, indicates that the client and server successfully exchanged the given (matching) test patterns using the ATM\_Echo RPC.

Begin processing configuration file "*name*"

This message, which appears in the trace file when the TR\_VERBOSE bit in the trace mask is set, confirms that the Client is reading the specified text-based configuration file.

Client exit code: *ddd*[*nnn*] - *exit\_code\_text*  
[(Client will try to recover from this error)]

This message appears any time a Client command completes. On UNIX, the 8-bit exit code is shown only if it differs from the actual exit code. The daemon uses the actual exit codes. You only need to deal with 8-bit exit codes in scripts that control the running of dbutility.

In the case of the command line Client (dbutility) a few exit codes cause a `process` command to restart itself after a brief delay. When this happens you will see the additional text "Client will try to recover from this error".

These exit codes include DBM\_AUD\_EOF (9), DBM\_WRONG\_ABSN (92), DBM\_BLOCKTOOLONG (1179), DBM\_LOC\_MISMATCH (33) and DBM\_ABSNMISMATCH (1180).

Additionally, if the client is terminated because a relational database deadlock was detected, dbutility will also attempt to recover by restarting after a brief delay.

Clustered System (*nnn* nodes)

(Windows only) This message, which is only written to the log file when running on a clustered system, displays the number of nodes in the cluster (typically 2).

Configuration information read from binary file "*name*"

This message, which appears in the trace file when the TR\_VERBOSE bit in the trace mask is set, confirms that the Client has successfully read the specified binary configuration file.

Connected to *host*, port *nnnn*

(Server connection only) This message, which appears in the trace file when the TR\_VERBOSE bit in the trace mask is set, indicates that the Client has successfully established a host connection, where *host* is the host name or IP address and *nnnn* is the TCP/IP port number.

Connecting to *host*, port *nnnn*

(Server connection only) Indicates that the Client is establishing a connection to the host, where *host* is the host name or IP address and *nnnn* is the TCP/IP port number.

Current date is: *Day Mon dd, yyyy*

This line appears, following a line of dashes, in the log file section for every Client run to identify the date on which the Client run was made. This line is also logged when the date changes after midnight, as all log messages only have a time prefix.

Databridge Client, Version *vvv* (*nn*-bit) {ODBC | OCI}/{SQLServer | Oracle} [*vv*/OS]

The version string *vvv* (e.g., 6.6.0.000) identifies the Databridge Client that you are running. The value of *nn* (64) identifies the CPU type. ODBC and OCI are the APIs used to access the relational database (SQL Server or Oracle). For UNIX platforms, the Oracle version for which the client is built (e.g., 11g) and the operating system (e.g., Linux) is also listed.

Databridge Configuration Server, Version *vvv* (*nn*-bit) {ODBC | OCI}/{SQLServer | Oracle} [*vv*/OS]

The version string *vvv* (e.g., 6.6.0.000) identifies the secondary Databridge Client (DBCIntCfgServer) that you are running. The value of *nn* (64) identifies the CPU type. ODBC and OCI are the APIs used to access the relational database (SQL Server or Oracle). For UNIX platforms, the Oracle version for which the client is built (e.g., 11g) and the operating system (e.g., Linux) is also listed. This program provides database access to the Client Console and the supports the Client Configurator. Unless you use the Client Configurator, this program only runs for brief periods of time and shuts down automatically after a minute of inactivity.

DBEngine Version: *version\_string*

(Server connection only) This message, which is only written to the log file, shows the version of the Databridge Engine being used on the mainframe. It is only present in commands that connect to the Databridge Server (or Enterprise Server).

DBEnterprise Version: *version\_string*

(Server connection only) This message, which is only written to the log file, shows the version of the Databridge Enterprise Server being used. It is only present in commands that connect to the Databridge Enterprise Server.

DBServer Task Number: *nnnn*

(Server connection only) This message, which is only written to the log file, shows the DBServer task number on the MCP system.

DBServer Version: *version\_string*

(Server connection only) This message, which is written to the log file shows the version of DBServer used. It is present in commands that connect to the Databridge Server (or Enterprise Server).

DBSupport Title: *file\_title*

(Server connection only) This message, which is only written to the log file, shows the file title of the Databridge Support Library being used on the mainframe (for example "(DB66)OBJECT/DATABRIDGE/SUPPORT/DEMODB ON DPACK"). It is only present in commands that connect to the Databridge Server (or Enterprise Server).

DBSupport Version: *version\_string*

(Server connection only) This message, which is only written to the log file, shows the version of the Databridge Support Library being used on the mainframe. It is only present in commands that connect to the Databridge Server (or Enterprise Server).

Disconnecting and restarting transport initialization

(Server connection only) This message, which only appears in the trace file when the TR\_VERBOSE bit in the trace mask is set, indicates that the response to the ATM\_Echo remote procedure call (RPC) timed out. The Client recovers from this error by disconnecting from the server and trying to restart the connection process (up to three times).

DMSIISupport Title: *file\_title*

(Server connection only) This message, which is only written to the log file, shows the file title of the Databridge DMSII Support Library being used on the mainframe (for example "(DB66)OBJECT/DATABRIDGE/DMSIISUPPORT/DEMODB ON DPACK"). It is only present in commands that connect to the Databridge Server (or Enterprise Server).

DMSIISupport Version: *version\_string*

(Server connection only) This message, which is only written to the log file, shows the version of the Databridge DMSII Support Library being used on the mainframe. It is only present in commands that connect to the Databridge Server (or Enterprise Server).

End processing configuration file "*name*"

This message, which only appears in the trace file when the TR\_VERBOSE bit in the trace mask is set, confirms that the Client has processed the specified text-based configuration file.

Filter: *filter\_name*

(Server connection only) This message, which is only written to the log file, shows the name of the FILTER being used in the Support Library on the mainframe. It is only present in commands that connect to the Databridge Server (or Enterprise Server).

Flushing garbage data

(Server connection only) This message only appears in the trace file when the TR\_VERBOSE bit in the trace mask is set. To verify that the transport layer works correctly, the Client exchanges a test pattern with DBServer. If the Client receives a response that has an incorrect pattern or length, the Client attempts to recover from this situation and returns this message. You should not see this message under normal circumstances.

Negotiated protocol level = *number*, Host version = *major\_vers.minor\_vers*

(Server connection only) This message, which is only written to the log file, shows the protocol level that the Client and the server use, which is the lesser of the client and server protocol levels (the 6.6 release uses a protocol level of 33). The second part of the message contains the major and minor version numbers of the server (e.g., 6.6).

ODBC driver: "*name*", version = *vv*

This message, which only applies to the SQL Server Client, identifies the name and the version of the ODBC driver used.

Oracle database name: *name*

This line is written to the log file only when the database parameter in the Client configuration file has no assigned value and the command line `-D` option has not been used to specify the database name. It indicates that the default database *name* is used. This is not typically done in a production environment, but it could be done when using Oracle Express to evaluate the product. The preferred way of doing things is to create an entry for the database in the file `"tnsnames.ora"` in the `"network/admin"` folder in the Oracle files.

Oracle version: *major\_vers.minor\_vers*

This line, which is only written to the log file, displays the major and minor version of the Oracle database you are using (e.g., 11.2). This message is present for all commands that connect to the Oracle database.

OS version: *version\_string*

This line, which is only written to the log file and is only applicable to UNIX clients, displays the Operating System version string (e.g., 6.1 on an AIX platform). This message is present for all commands that connect to the relational database. This information is crucial to support to determine if you are using a supported platform. For a complete list of supported platforms and system requirements, see the *Databridge Installation Guide*.

OS: Windows [Server] *version\_name*

This line, which is only written to the log file, displays the name of the Windows operating system. For servers you get the word "Server" followed by one of the following: 2008, 2008 R2, 2012, 2012 R2 or 2016. For desktop systems *version\_name* is one of the following: Vista, 7, 8, 8.1 or 10.

Retrying ATMEcho RPC after flushing input

(Server connection only) This message, which only appears in the trace file when the `TR_VERBOSE` bit in the trace mask is set, indicates that the Client received a response of the wrong length from DBServer and is retrying the `ATM_Echo` RPC.

Server communications initialization complete

(Server connection only) This message, which only appears in the trace file when the `TR_VERBOSE` bit in the trace mask is set, indicates that DBServer or Enterprise Server has successfully executed the `DBINITIALIZE` RPC.

SQL Server version: *major\_vers.minor\_vers*

This line, which is only written to the log file, displays the major and minor version of the Microsoft SQL Server database you are using. Note that SQL Server 2008 is version 10.0, SQL Server 2008 R2 is version 10.5, SQL Server 2012 is version 11.0, SQL Server 2014 is version 12.0, SQL Server 2016 is version 13.0, and SQL Server 2017 is 14.0. This message is present for all commands that connect to the SQL Server database.

System: *host\_system\_desc*

(Server connection only) This message, which is only written to the log file, shows information about the type of Unisys mainframe being used (for example, System: CS790 :7003 SSR 56.189.8111). It is only present in commands that connect to the Databridge Server (or DBEnterprise Server).

## Configure Command Messages

The following messages appear in response to the Databridge Client `configure` command.

"locks" directory created

This message appears when the `configure` command creates the locks subdirectory in the global working directory. The global working directory and the locks directory are created only if they do not already exist.

"name" sub-directory created

This message appears when subdirectories for the client's working directory are created. These subdirectories are `config`, `logs`, `dbscripts`, `discards`, and `scripts`.

Beginning Databridge Client configuration

This message indicates that the process of creating the control tables and their indexes has begun.

Client configuration file "dbridge.cfg" created in "config" sub-directory

This message appears when the Client creates the binary configuration file `dbridge.cfg` in the `config` subdirectory. If the directory already exists and contains an old configuration file, this file will be used instead.

Creating control table *name*

This message indicates that the specified control table and its associated index are being created in the relational database.

Databridge Client configuration completed

This message indicates that the `dbutility configure` command is complete and that all of the empty control tables were successfully created in the relational database.

Dropping control table *name*

This message indicates that the specified control table is being dropped from the relational database. When a Client control table is found to exist, this command will drop it before creating it again. This message appears when you have previously executed a `dbutility configure` command and for some reason are executing it again using the `-u` option.

Generating control tables...

This message indicates that the Databridge Client is creating the control tables.

Working directory "*path*" created

This message appears when the command creates the global working directory for the Client (that is, the working directory for the service). This directory and the locks subdirectory are created only if they do not already exist.

## Define Command Messages

The following messages appear in response to the Databridge Client `define` command.

"locks" directory created

This message appears when the `define` command creates the locks subdirectory in the global working directory. The global working directory and the locks directory are created only if they do not already exist.



"name" sub-directory created

This message appears for each subdirectory of the Client's working directory that is created. These subdirectories are config, logs, dbscripts, discards, and scripts.

Beginning New DataSource definition

The data source with the host name and port number specified on the command line is being defined. The `dbutility define` command will get an error if the specified data source is present in the control tables. To resolve this error, use the `-u` option to force delete old entries in the Client control tables.

Client configuration file "dbridge.cfg" created in "config" sub-directory

This message appears when the Databridge Client creates the config subdirectory and a binary configuration file, `dbridge.cfg`, in that directory. If the config directory already exists, the existing configuration file is used.

Creating file "source\_NullRec.dat"

The Client is creating the file, `source_NullRec.dat` to hold the NULL VALUES for data set records from the specified data source. `source` is the data source specified in the `data_source` column of the corresponding DATASOURCES Client control table.

DataSet *name* is a global data set, active column set to 0

The Client automatically disables cloning for the indicated data set. Normally, the data in the global data set is not cloned because it is not very useful. If you need to clone this data set, simply set `active = 1` in the corresponding row of the DATASETS Client control table using a user script or the Client Configurator. If you are not using the Client Configurator, you will need to rerun the `define` command with the `-u` option (or run a `redefine` command with the `-R` option) to make the change take effect.

DataSet *name* is a restart data set, active column set to 0

The Client automatically disables cloning for the DMSII restart data set.

DataSource definition completed

This message indicates that the `define` command is complete. This means that a row with the data source name, host name, and port number has been added to the DATASOURCES control table. In addition, a row for each data set has been created in the DATASETS control table. The DMSII layout information has been downloaded to the DMS\_ITEMS control tables; and the corresponding relational database table layout information has been created in the DATATABLES and DATAITEMS control tables.

```
DB_Info: update_level = ddd, update_ts = timestamp, highest_strnum = ddd
DB_Info: database_ts = timestamp, database_name = name
[DB_Info: OptionFlags=options]
```

These messages are only written to the log file. The first line provides the database update level and timestamp, and the highest structure number in the database for the data source being accessed. The second line provides the database timestamp and the database name for the data source being accessed. A third line of comma-separated options appears when any of these options are true: `IndependentTrans` (`INDEPENDENTTRANS` is set for the DMSII database); `AccessActive` (`READ ACTIVE AUDIT` is set in the Engine Control File); `RDB` (the DMSII database is an RDB secondary database); `FileXtract` (the data source is a FileXtract file); and `LINKS` (`LINKS` is set to `TRUE` in the Engine Control File).

Defining table entries for DataSet *name*[*rectype*] (*struct\_number*)

The Client control table entries for the specified data set are being defined. */rectype* appears only for variable-format data set records that have a nonzero record type (they contain a variable part). For more information, see "Variable-Format Data Sets" in the *Databridge Client Administrator's Guide*. *struct\_number* is the DMSII structure number of the data set. The Databridge Engine processes data sets in structure number order; the structure number is an indication of how close the `define` command is to completing.

Inserting data into control tables...

This message indicates that data is being inserted into the control tables.

Launching `makefilter` utility to create the binary filter file "dbfilter.cfg" in the config directory

This message indicates that the client is launching the `makefilter` utility to compile the filter "dbfilter.txt" that was found in the config subdirectory. The `makefilter` output is written to a separate log file named "*prefixflt\_yyyyymmdd.log*" (where *prefix* is the prefix used for the client log file and defaults to "db").

Loading control tables for *datasource*

This message appears at the beginning of the second phase of the `define` command, when the Client reloads the control tables in order to pick up updates that result from the running of user scripts that alter the DMSII layout (`script.user_layout.primary_tablename`).

Mapping table entries for DataSet *name[/rectype]* (*struct\_number*)

This message, which normally only appears in the trace file when the `TR_VERBOSE` bit in the trace mask is set, indicates that the DMSII layout for the data set in question is being mapped to the corresponding relational database tables. It only appears in the log file if the `status_bits` column in the corresponding `DATASETS` table entry has the `DS_Needs_Mapping` bit (4) set.

Rows updated = *ccc*

This message, which only appears in the log file when the `-v` option is enabled, shows the row count for all SQL statements that are executed when processing user scripts. A value of 0 is usually an indication that the user script is in error. At this point, it's a good idea to rerun the command with user script tracing and log output tracing enabled (that is, `-t 2049`). Log output tracing creates a trace file with both row counts and the SQL statement, which can be hard to match up otherwise.

Running script *script\_file\_spec*

This message is a confirmation that the Client is running the specified script. For each SQL statement in the script, when the `-v` option is enabled, the Client writes the number of rows that have been changed to the log file. See the preceding message, "Rows updated = *ccc*", for more information.

Working directory "*path*" created

This message appears when the `define` command needs to create the global working directory for the Client (that is, the working directory for the service). This directory and the locks subdirectory are created only if they do not already exist.

## Redefine Command Messages

The following messages appear in response to the Databridge Client `redefine` command. These messages can also appear when running the Client Configurator from the Client Console.

#### Beginning updates of DataSource definitions

This message indicates that the data source definitions are being recreated. Old Client control table entries that correspond to data sets which need to be redefined (for the specified data source) are deleted first. However, any customization that was done to them using the Client Configurator will be preserved. In the case of dbutility this is normally achieved by using user scripts, unless the configuration parameter `use_dbconfig` is set to True.

#### Command returned a status of *nnn*

This message, which only appears when using the Client Configurator, shows the result of the Compare Layouts operation, which determines if the relational database layout has changed as a result of the data source being redefined. Possible values for *nnn*:

- ◆ 0 indicates that the layouts are unchanged
- ◆ 2032 indicates that a `generate` command is required (most likely because there are some datasets that have tables that need to be cloned or recloned)
- ◆ 2033 indicates that a `reorg` command is required, as the changes can be made without having to reclone anything.
- ◆ Any other exit status indicates that an error has occurred while comparing the layouts. In this case, there should be earlier error messages you need to look at.

The same message is also used to indicate that a remote procedure call by the Client Configurator returned a non-zero status, which means that the RPC encountered an error.

#### Command returned a status of *nnn (text)*

This message, which only appears when using the Client Console, shows the result of a Define/Redefine command issued by the Client Console. The message text gives a brief explanation of the error and can be one of the following:

- ◆ No Further Action Required - processing can continue - In the case of an existing data source this indicates that the `redefine` command does not require that a `reorg` or `generate` command be issued.
- ◆ You need to run a Reorganize command - In the case of an existing data source this indicates that the `redefine` command has found differences between the old and new database layouts and that you need to run a `Reorganize` command to alter the tables.
- ◆ You need to run a Generate command - This indicates that the scripts in the `dbscripts` directory need to be generated before you can run a `process` command.

#### Comparing old and new relational database layouts

This message, which only appears when using the Client Configurator, indicates that old and new relational database layouts are being compared. The results of this operation determine if the data source needs any special attention prior to resuming normal operations, such as running a `generate` or `reorg` command.

#### Creating file "*datasource\_NullRec.dat*"

This message indicates that the Client is creating the file, `datasource_NullRec.dat` to hold the NULL VALUES for data set records from the specified data source. `datasource` is the source specified in the `data_source` column of the corresponding DATASOURCES Client control table. A `redefine` command with the `-R` option ("Redefine All Data Sets" when using the Client Console) recreates the Null Record file. If you accidentally delete this file, this how you go about recreating it.

DataSet *name[/rectype]* did not previously exist -- you will need to run a generate command

This message indicates that a new data set was added to the DMSII DASDL since the last time a `define` command or a `redefine` command was executed and is a reminder that you need to execute a `generate` command before cloning the new data set. This message only appears when the parameter `suppress_new_datasets` is set to `False`.

DataSet *name[/rectype]* did not previously exist, defined with the active column set to 0

This message indicates that a new data set was added to the DMSII DASDL since the last time a `define` command or a `redefine` command was executed. This message only appears if the parameter `suppress_new_datasets` is set to `True`. In this case, the active column for the new data set is set to 0 in the DATASETS table.

DataSet *name[/rectype]* no longer exists

The specified data set was deleted from the DMSII DASDL since the last time a `define` or `redefine` command was executed.

DataSet *name[/rectype]* unaffected by reorg ds\_mode is *n*[, however a generate command is required as `status_bits=ssss`]

The layout of the relational database table mapped from the specified data set was not affected by the database reorganization. Therefore, the specified data set does *not* require recloning. In the unusual case where the command requires that you execute a `generate` command, the full message is shown.

DataSet *name[/rectype]* will be re-defined

This message, which only appears in the trace file when the `TR_VERBOSE` bit in the trace mask is set, indicates that the specified data set will be redefined

```
DB_Info: update_level = ddd, update_ts = timestamp, highest_strnum = ddd
DB_Info: database_ts = timestamp, database_name = name
[DB_Info: OptionFlags=options]
```

These messages are only written to the log file. The first line shows the database update level, the database update timestamp and the highest structure number in the database for the datassource that is being accessed. The second line shows the database timestamp and the database name for the data source that is being accessed. The third line lists comma-separated options and only appears when any of the following is true: `IndependentTrans` (`INDEPENDENTTRANS` is set for the DMSII database), `AccessActive` (`READ ACTIVE AUDIT` is set in the Engine Control File), `DBPlus` (`DBPlus` is being used by the Engine), `RDB` (the DMSII database is an RDB secondary database), `FileXtract` (the data source is a `FileXtract` file) and `LINKS` (`LINKS` is set to `TRUE` in the Engine Control File).

Defining table entries for DataSet *name[/rectype]* (*struct\_number*)

The client control table entries for the specified data set are being defined. */rectype* appears only for variable-format data set records that have a nonzero record type. (They contain a variable part.) *struct\_number* is the DMSII structure number of the data set. Because the Databridge Engine processes data sets in structure number order, the structure number is an indication of how close the `define` command is to completing.

Format level change *mmm* -> *nnn* detected for DataSet *name[/rectype]*

This message, which is printed at the start of a `redefine` command, indicates that the format level for the given data set has changed. Every time a data set is changed in the DASDL, the format level for the data set is set to the database update level when the DASDL is recompiled. A format level change is an indication that the data set was affected by a DMSII structural reorganization.

Item count change *mmm* -> *nnn* detected for DataSet *name[/rectype]*

This message, which is printed at the start of the `redefine` command, indicates that the number of items in the data set has changed. This could be the result of a filler substitution reorganization or a change in the column filtering specified in GenFormat.

Loading control tables for *datasource*

The `redefine` command shows this message every time it loads the Client control tables. The `redefine` command always starts by loading the Client control tables. The Client control tables are also reloaded after running user scripts, at the end of the two main phases of the command, in order to pick up updates that result from the running of these scripts.

Mapping table entries for DataSet *name[/rectype]* (*struct\_number*)

This message, which normally only appears in the trace file when the TR\_VERBOSE bit in the trace mask is set, indicates that the DMSII layout for the data set in question is being mapped to the corresponding relational database tables. It only appears in the log file if the status\_bits column in the corresponding DATASETS table entry has the DS\_Needs\_Mapping bit (4) set.

No Further Action Required - Processing can continue

This message, which only applies to the command line Client dbutility, is a confirmation that the `redefine` command did not detect any layout changes and you can resume processing audit files.

Redefine DataSet *name[/rectype]*: DSOPT\_NoStoredProcs option bit was changed from *aa* to *bb*

This message indicates that the use of stored procedures for the data set in question has changed. This happens when you change the bit by either changing the `use_stored_procs` configuration parameter or by using a user script. The Client will ask you to run a `reorg` command to fix things up. The `reorg` command will create a new set of scripts and it will refresh the stored procedures for the data set. This means that when you go from using stored procedures to not using them, they will be dropped and conversely if you go from not using stored procedures to using them they will be created.

Redefine DataSet *name[/rectype]*: index name for table '*name*' was changed from '*name1*' to '*name2*'

This message indicates that the name of the index for the given table has changed. The client simply renames the index using a `reorg` command to avoid problems at a later time.

Redefine DataSet *name[/rectype]*: index type for table '*name*' was changed from *index\_type1* to *index\_type2*

This message indicates that index type for the given table has changed (for example, from a unique index to a primary key). The client drops the old index and creates the new index using a `reorg` command.

Redefine DataSet *name[/rectype]*: Table '*name*' no longer being used as a result of the DMSII `reorg` - Run the script '*script.drop.name*' to drop the table and its stored procedures

As a result of the DMSII reorganization of the specified data set, the specified table was removed from the DATATABLES and DATAITEMS entries mapped from this data set. The table is not dropped. To drop the table and the stored procedures associated with it, execute the script noted in the message.

```
Redefine DataSet name[/rectype]: Table 'name' was added [--active column set to 0 in DATATABLES]
```

As a result of the DMSII reorganization of the specified data set, the specified table was added to the DATATABLES and DATAITEMS entries mapped from this data set. If the parameter `suppress_new_columns` is set to True, the Databridge Client sets the active column of this entry (in DATATABLES) to 0.

```
Redefine Table 'name': Column 'name' (Item# number), changed from dec_type(p1[,s1]) to dec_type(p2[,s2])
```

As a result of the DMSII reorganization, the data type of the specified column has changed. `dec_type` indicates the values of the `sql_type` column of the corresponding DATAITEMS table entries. `p1` and `p2` indicate the values of the `sql_length` columns. `s1` and `s2` indicate the values of the `sql_scale` column if applicable for the specified `sql_type`.

```
Redefine Table 'name': Column 'name' (Item# number) dms_subtype value changed from mmm to nnn
```

This indicates that the `dms_subtype` for the given column changed. If this is a date this means that the format of the DMSII data is now different. This usually means that you will need to reclone the data set.

```
Redefine Table 'name': Column 'name' (Item# number) item_key value changed from n(key#=k1) to m(key#=k2)
```

This indicated that the order of the columns in the index have changed. The numbers in parentheses are the actual positions of the columns within the index. As long as they're the same, the client ignores any changes in the `item_key` values.

```
Redefine Table 'name': Column 'name' (Item# number) masking_info value changed from 0xhhhh to 0xhhhh
```

As a result of a DMSII reorganization (or a change in customization for the column) the state of the masking for the column in question was changed. This will be handled by the ensuing `reorg` command which will ALTER the table.

```
Redefine Table 'name': Column 'name' (Item# number) NULL attribute changed from old_val to new_val
```

As a result of a DMSII reorganization, the NULL item attribute changed. In addition, the message shows both the old and new values. The client alters the column using the `reorg` command, which runs the `reorg` scripts created by the `redefine` command.

```
Redefine Table 'name': Column 'name' (Item# number) SQL length value changed from old_length to new_length
```

As a result of a DMSII reorganization, the SQL length for the specified column in the specified table changed. In addition, the message shows both the old and new values. The client alters the column using the `reorg` command, which runs the `reorg` scripts created by the `redefine` command.

```
Redefine Table 'name': Column 'name' (Item# number) SQL type value changed from old_type to new_type
```

As a result of a DMSII reorganization, the SQL type for the specified column in the specified table changed. In addition, the message shows both the old and new values. The client alters the column using the `reorg` command, which runs the reorg scripts created by the `redefine` command provided that the type of transformation is allowed. For example, if a column with a data type of `int` changes to `date`, the data set must be recloned; the `alter` command does not support this type of transformation.

```
Redefine Table 'name': Column 'name' (Item# number) was added [--active column set to 0 in DATAITEMS]
```

As a result of a DMSII reorganization, the specified column was added to the specified table. If the parameter `suppress_new_columns` is set to `True`, the Client sets the active column of this entry (in `DATAITEMS`) to 0. If the parameter `suppress_new_columns` is set to `False`, the client adds the column to the table using the `reorg` command, which executes a script that does the actual `alter` command.

```
Redefine Table 'name': Column 'name' (Item# number) was deleted
```

As a result of a DMSII reorganization the specified column in the specified table was removed. The client drops the column from the table using the `reorg` command, which executes a script that does the actual `alter` command. If you want to keep the column, modify the script not to drop the column. Make sure that the column has the `NULL` attribute or has a `DEFAULT` defined; the client will not provide a value for this column. Otherwise, the update will fail.

```
Redefine Table 'name': Column 'name' (Item# number) will be handled as a nullable key
```

This message, which is only applicable to MISER databases, indicates that a MISER date, which is a key, has been encountered when the parameter `use_nullable_dates` is set to `True`. The client will generate special code for the update stored procedure to handle the case when the value of the key item in question is `NULL`.

**NOTE:** This code is limited to one nullable date in the index of the table. If you have more, you must use a different index.

```
Restoring user changes to DATATABLES and DATAITEMS tables
```

This message only appears when the `redefine` command runs in Client Configurator compatible mode (that is, the configuration parameter `use_dbconfig` is set to `True`). If the data source was not created using the Client Configurator, you must run the `dbscriptfixup` program before you can enable the parameter `use_dbconfig`.

When run in this mode, the `redefine` command does not use user scripts. Instead, it restores the changes from the old copy of the control table. This message, which only appears in the trace file when the `TR_VERBOSE` bit in the trace mask is set, indicates that the customization changes for the relational database tables and their columns are being restored from the old copy of the control tables.

```
Restoring user changes to DMS_ITEMS table
```

This message, which only appears in the trace file when the `TR_VERBOSE` bit in the trace mask is set, indicates that the customization changes for the `DMS_ITEMS` entries are being restored from the old copy of the control tables. See the preceding message for details.

```
Rows updated = ccc
```

This message, which only appears if the `-v` option is enabled, shows the row count for all SQL statements that are executed when processing user scripts. A value of 0 is usually indicates that the user script is in error. When this occurs, it's a good idea to rerun the command with the user script tracing and log output tracing enabled (that is, `-t 2049`). This creates a trace that provides both the SQL statement and the row counts, which are otherwise hard to match up.

Running script "*script\_file\_spec*"

This message confirms that the Client is running the specified script. For each SQL statement in the script, if the `-v` option is enabled, the Client writes the number of rows that have been changed to the log file. For details, see the preceding message.

Update of definitions of DataSets completed

This message indicates that the `redefine` command is complete. As a result, the DMSII layout that the corresponding relational database table layout information has been updated in the Client control tables. If the layout has changed, the scripts that the `reorg` command executes are written to the working directory for the data source. Additionally, the original control tables are saved in the unload file named *source\_reorg\_nnn.cct*, where *source* is the data source name and *nnn* is the original update level of the DMSII database. If the command fails because of a bad user script, you can reload the control tables from this file and re-execute the command after fixing the bad user script.

You must also run a `generate` command for DataSet *name[/rectype]* which is to be re-cloned

This message is a reminder that the table layout for the data set in question has changed and a `generate` command must be run before the data set can be re-cloned. Failure to do so will simply result in the process command failing. This message sometimes follows the message "WARNING: AA values for DataSet *name[/rectype]* are no longer valid, `ds_mode` set to 0 to force a re-clone".

## Generate Command Messages

The following messages appear in response to the Databridge Client `generate` command.

Beginning script generation

This message indicates that the scripts for creating, populating, and updating the Databridge data tables in the relational database are being created in the `dbscripts` subdirectory.

Creating scripts for '*tabname*'

This message indicates that scripts for each table that will be stored in the relational database are being created. This status message appears one time for each DMSII data set you are cloning.

Generate command found nothing to do -- all scripts are current

This message indicates that the `generate` command did not find any data sets whose `DS_Needs_Generating` bit in the `status_bits` column of the DATASETS Client control table were set. The program suppresses unnecessary script generation to avoid overwriting any changes users might have made to these scripts. To disable this safeguard, use the `-u` option for the `generate` command; or from the Client console, select the **Generate All Scripts** command.

Loading control tables for *datasource*

This message indicates that the DMSII and relational database layout information is being loaded into memory from the Client control tables.



Script generation completed

This message indicates that the `generate` command is complete and that the necessary script files have been written to the `dbscripts` directory.

## Reorg Command Messages

The following messages appear in response to the Databridge Client `reorg` command.

Clearing DataSet `name[/rectype]` records

This message indicates that the cleanup scripts is being run for the all the tables of the specified multisource data set.

Creating history table '`name`'

This message indicates that a history table is being created because the configuration parameter `enable_dynamic_hist` is set to True.

Creating index '`name`' for history table '`name`'

This message indicates that the index for a dynamically-added history table is being created.

Creating index '`name`' for table '`name`'

This message indicates that the `reorg` command is creating the index for a table whose reorganization requires the index to be dropped and recreated.

Creating scripts for '`tablename`'

This message is printed as a result of the `reorg` command executing a `generate` command before it does anything else.

Dataset `name[/rectype]` successfully reorganized

This message is printed after a data set is successfully reorganized. It is meant to show the progress of the command that might sometime take a very long time to complete.

DataSource `name` successfully reorganized

This message is printed at the end of a `reorg` command to indicate that the command completed successfully. It is mainly meant to provide some feedback when the command is executed from the Client Console.

Dropping index for table '`name`'

This message indicates that the `reorg` command is dropping the index for a table whose reorganization requires the index to be dropped and recreated.

Loading control tables for `datasource`

This message indicates that the DATASETS table and relational database layout information is being loaded into memory from the client control tables.

Multisource DataSet `name[/rectype]` ds\_mode is 35  
The multisourced table has been dropped as there are no records left  
Set ds\_mode to 0 for all of the sources to reclone

These messages indicate that a multisource data set was dropped and that the other data source has also been dropped. When the reorganization of both data sources is complete, set the `ds_mode` to 0 to reclone it. See the next message for more details.

```
Multisource DataSet name[/rectype] ds_mode is 35  
Set ds_mode to 0 when all sources have been reorganized
```

These messages indicate that a multisource data set was dropped and that the other data source must be reorganized before further action can occur. When the reorganization of that data source is complete, set the `ds_mode` to 0 to reclone it.

Multisourced data sources get their inputs from two structurally identical DMSII databases (same DASDL) that reside on different systems (for example, two different branches of the company, each with their own database). The data is stored in the same tables in the relational database, using the `source_id` columns in all the tables to keep track of where the records originated.

When reorganizing a multisourced data source, the changes must be completed in both data sources before you can alter the tables. The data source that is reorganized first is placed in a waiting state until the remaining data source is reorganized. If the `reorg` command determines that the table contains no records that come from any of the other multisourced data sources, it issues this message to indicate that it is time to reclone the table.

```
Reorganizing DataSet name[/rectype]
```

This message is printed to the log file when a data set whose mode is 31 is found. When the reorganization is completed, the message "Dataset *name*[/rectype] successfully reorganized" is shown.

```
Stored procedures for all tables of DataSet name[/rectype] successfully refreshed
```

This message appears as a result of the `reorg` command executing the code that refreshed the stored procedures for the tables of a data set.

## Process and Clone Commands Messages

The following messages appear in response to the Databridge Client `process` and `clone` commands.

```
ABORT command initiated by TERM signal
```

This message, which is limited to UNIX client, indicates that a kill command was used to generate a SIGTERM signal that the client is responding to. The Client treats this signal exactly like a command line console QUIT NOW command. This is particularly useful, if you are running the Client as a background run.

```
Attempting to clear duplicate records for table 'name'
```

This message indicates that following the failure to successfully create an index for a table during the data extraction phase, the program will attempt to run the script "`script.clduprecs.tablename`" to remove duplicate records from the table. This situation can occur if the Databridge Engine sees the same record twice during data extraction. This is much more likely in the case of compact data sets where records can move around in the data set when their sizes change.

Aux STMT Statistics:

```
Configured_max = nnn, Max_used = mmm, Recycled_stmt_cnt = rrr  
STMT reuse stats: min_sql_ops = nnn, max_sql_ops = mmm, avg_sql_ops = rrr.rr  
STMTs never reused = nnn, min_sql_ops = mmm, max_sql_ops = ddd, avg_sql_ops =  
rrr.rr
```

This message appears after the cumulative statistics at the end of the run. It shows the auxiliary statements used by the Client.

The first line shows *nnn*, the value of the `aux_stmts` parameter in the Client configuration file, and *mmm*, the maximum number of statements that were used. The recycled statement count, *rrr*, indicates how many statements were reused. If the *rrr* value is high (or if the *nnn* and *mmm* values are the same) you may not have enough statements configured. If you change these values, keep in mind that higher values can result in better performance but will require more memory.

The second line represents the number of SQL statements that were executed using a given auxiliary statement. The minimum, maximum and average values are shown. The minimum value is typically 1, if you have tables that are very rarely updated. A high value for the maximum is encouraging, but it can be misleading if you have a small number of tables that get updated a lot. If the average value is high, this indicates that you have enough auxiliary statements. Any statement that is reused will run much faster than one that executes for the first time, as the first execution requires additional I/O. This is particularly visible in the Oracle client, where the speed up is quite visible.

The third line represents the minimum, maximum, and average number of SQL operations that were executed by these SQL statements and the number of SQL statements that have not been reused. If this number is close to the number of configured auxiliary statements, you might benefit from increasing the value of the configuration parameter `aux_stmts`. This would allow more SQL operations to re-use statements, thereby improving performance.

```
Begin populating/updating database [from AFN=afn, ABSN=bsn, SEG=seg, INX=inx,  
DMSII Time=timestamp]
```

This message appears at the start of a `process` or `clone` command after all the data sets have been successfully selected. The absence of any audit file information indicates that all of the selected data sets need to be cloned and that the Databridge data tables have been successfully created. The audit file location information indicates that a process command has found at least one data set ready to receive DMSII updates.

```
BI image for update to table 'name' is now filtered, deleting old image; Keys:  
column_name = value, ...
```

This message, which appears in the trace file when the `TR_VERBOSE` bit in the trace mask is set, indicates that row in question, that was not previously filtered, now satisfies the filtering condition and needs to be removed from the corresponding table. The client automatically deletes this row from the table in this situation.

```
BI image for update to table 'name' was previously filtered, inserting new image;  
Keys: column_name = value, ...
```

This message, which appears in the trace file when the `TR_VERBOSE` bit in the trace mask is set, indicates that row in question, that was previously filtered, no longer satisfies the filtering condition and needs to be stored in the corresponding table. The client automatically inserts this row into the table in this situation.

```
Build_BCP_Record: table='name', record filtered for name1 = occ
```

This message applies to the SQL Server Client during data extraction, and indicates when an OCCURS table filter is present. The message indicates that the specified occurrence of the item (or GROUP) is being suppressed as it meets the filtering conditions.

Bulk loader parameter "max\_temp\_storage" = *mmm* MB

(Windows only) This message, which is only written to the log file when one or more data sets are to be cloned, records the value of the `max_temp_storage` parameter in the Client configuration file. This provides readily available information for analyzing and resolving a slow clone. You should use a value of at least 400 MB for this parameter.

Bulk load count verification for table '*name*' complete: *number* rows

This message only appears when the configuration parameter `verify_bulk_load` is set to 1 or 2 and the number of records from the table (from the relational database) is equal to the number of loaded records.

Bulk\_loader thread no longer hung, main thread resuming

(Windows only) The bulk loader thread, which had fallen so far behind the main thread that it caused the main thread to block, has caught up and allowed the main thread to unblock and start running again. If this situation occurs again, you must investigate why the bulk loader is running so slowly. If you are using a remote connection over Oracle, try increasing `sql_bindsizes` to 1 MB.

Cleaning up table '*name*' [fully]

This message indicates that the Client is deleting selective records from the specified table at the beginning of the data extraction phase, instead of dropping the table and recreating it. This action is taken only in special cases, such as when deleted records are preserved. Another case is when a table that gets its input from more than one data set is partially reloaded. The presence of the word "fully" indicates that the script "`script.cleanup2.table`" is being run, as opposed to the script "`script.cleanup.table`".

Clear duplicate records script ran successfully

This message confirms that the script "`script.clrduprecs.tablename`" was successfully run. The purpose of this script is to delete records that appear multiple times in the table. The fixup process will reinsert the correct copy of these records. This situation is rare, but tends to happen when compact data sets are involved.

**CAUTION:** You should disable the use of this script when creating composite keys that you are unsure of, as running this script could end up deleting perfectly good data and forcing you to reload, instead of deleting duplicate records encountered during the data extraction. To disable the running of the script for a particular data set, you need to reset the bit `DSOPT_Clrdup_Recs` (32,768) in the `ds_options` column of the corresponding row in DATASETS Client control table.

Clone of selected DataSets completed

This message indicates that the clone command for the specified data sets completed successfully.

Closing file "`bcppipe.name[_number]`"

This message, which applies only to the SQL Server Client and appears in the trace file when the `TR_VERBOSE` bit in the trace mask is set, indicates that the specified temporary file has been successfully closed prior to being queued for bulk loading.

Closing file "`lpipe_number[_number].dat`"

This message, which applies only to the Oracle Client and appears in the trace file when the TR\_VERBOSE bit in the trace mask is set, indicates that the specified temporary file has been successfully closed prior to being queued for bulk loading.

Creating index '*name*' for table '*name*'

This message shows the progress of the data set being cloned and indicates that the Client is creating an index for the specified table.

Creating table '*name*'

This message, which shows the progress of the data set being cloned, indicates that the specified table is being created in the relational database.

Creating temp file "*name*" for table '*name*'

(Windows only) This message shows the progress of the clone during the data extraction phase and indicates that the Databridge Client is creating a temporary file named *lpipe\_tablename.dat*. This message is only printed when the first temporary file for the given table is created, unless the TR\_VERBOSE bit in the trace mask is set. In which case it is written to the trace file every time a temporary file is created.

Cumulative Statistics:

This message contains statistics for the entire run (compared to incremental statistics, which applies only to each individual audit file processed). This message is written to the log file at the end of the run. For a description of each field, see "Incremental Statistics" in this section.

Data extraction phase for table '*name*' complete, *num* DMSII records processed, *num* rows loaded [,*num* rows in error][,*num* rows discarded][,*num* rows filtered]

Indicates that the data extraction phase for the specified table is complete. It also reports the DMSII record count, the corresponding number of rows that were loaded into the relational database, the count of records that had data errors, the number of discarded records, which are placed into the program discard files (*tablename.bad*) in the discards subdirectory, and the number of filtered out records. If any of the last 3 counts are 0 they are omitted from the message.

Note that records discarded by the bulk loader are placed in different files in the discards subdirectory. The files are named *bcp.tablename.bad* and *sqlld.tablename.bad* for the SQL Server and Oracle clients respectively.

Data Extraction [Phase 1] Statistics:

This message precedes a list of data extraction statistics that are printed to the log file.

In the case of a MISER database, which has virtual data sets that get their input from more than one data set, the data extraction has two phases. The message at the end of the first phase of the data extraction is marked "Phase 1". The statistics at the end of the second phase include both phases.

Data Extracted *nnn.nn* KB in *sss.sss* secs, throughput = *ddd.dd* KB/sec,  
DMSII recs/sec = *rrr.rr*

*nnn.nn* represents the number of kilobytes of DMSII data received. *sss.sss* represents the elapsed time (in seconds). *ddd.dd* represents the corresponding throughput. This message is always followed by the message "Bytes Received *nnn.nn* KB in *sss.sss*secs , total throughput = *ddd.dd* KB/sec," which shows the total data throughput rather than the actual DMSII data throughput. *rrr.rr* represents the corresponding rate at which DMSII records were processed.

Bytes Received *nnn* KB in *sss.sss* secs, total throughput = *ddd.ddd* KB/sec

*nnn.nn* represents the number of kilobytes of total data received (including the packet headers and the non-data packets). *sss.sss* represents the elapsed time in seconds. *ddd.dd* represents the corresponding throughput.

DMSII Buffers Used = *dd* (configured\_max = *mm*)

*dd* is the actual number of DMSII buffers used and *mm* is the configured maximum. Unless you have DMSII links the value of *dd* will be 1. CAUTION: When you have DMSII links the data extraction will use more buffers, therefore you should not try to reduce the default setting as this can cause the clone to fail if it runs out of DMSII buffers.

TCP/IP time = *sss.sss* secs, (*dd.dd*% of total time)

*sss.sss* represents the amount of time (in seconds) that the program spent waiting for TCP/IP data to appear from the host. *dd.dd* is the corresponding percentage of total elapsed time.

SQL\_exec time = *sss.sss* secs, (*dd.dd*% of total time)

*sss.sss* is the amount of time (in seconds) that the main thread of the program spent waiting for the execution SQL statements to complete. *dd.dd* is the corresponding percentage of total elapsed time.

TXN\_exec time = *sss.sss* secs, (*dd.dd*% of total time)

*sss.sss* represents the amount of time (in seconds) the main thread of the program spent waiting for commits to complete. *dd.dd* is the corresponding percentage of total elapsed time.

File\_I/O time = *sss.sss* secs, (*dd.dd*% of total time)

This line indicates the amount of time that the main thread spent waiting for file I/O operations to complete. For Windows, this represents the I/O writing the temporary files. For UNIX, it represents the amount of time spent writing to the UNIX pipes. These times also include any blocking time when the bulk loader falls behind the main process. In all cases, *sss.sss* is the amount of time (in seconds) the program spent waiting on file I/O operations to complete. *dd.dd* is the corresponding percentage of total elapsed time.

Bulk\_load time = *sss.sss* secs, (*dd.dd*% of total time)

(Windows only) This line, which is limited to the single-thread case (*n\_update\_threads* = 0), indicates the amount of time that the main thread spent waiting for the bulk loader operations. The value *sss.sss* is the amount of time in seconds that the main thread spent waiting on resources tied up by the bulk loader thread and *dd.dd* is the corresponding percentage of total elapsed time. A high value of *Bulk\_load\_time* might be an indication that the value of *max\_temp\_storage* you are using is too small. For MISER databases setting *max\_temp\_storage* to 1 GB seems to work best.

```
buf_wait time = sss.sss secs, (dd.dd % of total time)
```

This line only applies to multi-threaded extracts. *sss.sss* represents the amount of time (in seconds) the main thread of the program spent waiting for a DMSII buffer to become available. *dd.dd* is the corresponding percentage of total elapsed time. DMSII buffers are used to hold the raw DMSII records while they are being processed by the client. In the case of multi-threaded extracts these buffers are placed on the various updater threads' work queues using working storage header blocks described below.

```
ws_wait time = sss.sss secs, (dd.dd % of total time)
```

This line only applies to multi-threaded extracts. *sss.sss* represents the amount of time (in seconds) that the main thread of the program spent waiting for a working storage header block to become available. *dd.dd* is the corresponding percentage of total elapsed time. Working storage header blocks are small blocks used to queue DMSII buffers, as they can be on multiple queues when a DMSII data set maps to multiple tables.

```
Inx_wait time = sss.sss secs, (dd.dd % of total time)
```

*sss.sss* is the amount of time (in seconds) that the main thread of the program spent waiting for the index creator thread to finish before it could start the fixup phase. *dd.dd* is the corresponding percentage of total elapsed time.

```
CPU/other time = sss.sss secs, (dd.dd% of total time)
```

*sss.sss* represents the amount of time (in seconds) the main thread of the program was not waiting for any of the above actions to complete. *dd.dd* is the corresponding percentage of total elapsed time. This metric is a derived value and it represents the time during which the thread was in a runnable state.

```
Server_Packet_Counts:  
CREATE:ddd DELETE:ddd MODIFY:ddd MOD_BI:ddd MOD_AI:ddd STATE:ddd  
LINK_BI:ddd LINK_AI:ddd BULKDEL:ddd DOC:ddd COMMIT:ddd ROLLBK:ddd
```

These values represent the counts of the various record types and commits received from the Databridge Server. During data extraction, CREATE represents the number of records extracted. When using DMSII links, LINK\_AI represents extracted data for the link images in the various records. STATE represents StateInfo records. All the remaining record types will be zero at the end of the data extraction.

```
Bulk-loader Thread Statistics:  
Load_time = sss.sss secs, (dd.dd% of total time)  
Idle_time = sss.sss secs, (dd.dd% of total time)  
Other_time = sss.sss secs, (dd.dd% of total time)
```

(Windows only) The value *sss.sss* in the first line is the amount of time (in seconds) that the bulk-loader thread spent waiting for the bulk loader executions to complete. *dd.dd* is the corresponding percentage of total elapsed time.

In the second line, *sss.sss* is the number of seconds that the thread spent waiting for something to do. *dd.dd* is the corresponding percentage of total elapsed time.

In the third line *sss.sss* is the amount of time (in seconds) that the thread was not waiting for any of the above actions to complete and *dd.dd* is the corresponding percentage of total elapsed time. This metric is a derived value and it represents the time during which the thread was in a runnable state.

```
Index Thread Statistics:
Load_time   = sss.sss secs, (dd.dd% of total time)
Idle_time   = sss.sss secs, (dd.dd% of total time)
Other_time  = sss.sss secs, (dd.dd% of total time)
```

The value *sss.sss* in the first line is the number of seconds that the index-creator thread spent waiting for the execution of the create index SQL statements to complete. *dd.dd* is the corresponding percentage of total elapsed time.

In the second line *sss.sss* is the amount of time (in seconds) that the thread spent waiting for something to do and *dd.dd* is the corresponding percentage of total elapsed time.

In the third line *sss.sss* is the amount of time (in seconds) that the thread was not waiting for any of the above actions to complete and *dd.dd* is the corresponding percentage of total elapsed time. This metric is a derived value and it represents the time during which the thread was in a runnable state.

```
[Data Errors: eeeee SQL rows discarded, ddddd SQL rows in error]
```

This line shows the number of records that had data errors or were discarded. It doesn't appear when both counts are zero.

```
[Filter: ddddd occurs tables records suppressed]
```

This line shows the number of records that were discarded from OCCURS tables because of filtering. It doesn't appear when the count is zero.

```
Database clone/update completed
```

This message appears when a `process` or `clone` command completes successfully. This message can also appear when the Databridge Engine has successfully read the available audit files but there was no updated data to pass on to the Client.

```
DataSet name[/rectype], mode 1->2, datasets_to_fixup = nnn
```

This message, which appears in the trace file when the `TR_VERBOSE` bit in the trace mask is set, indicates that a State Information record with a mode of 2 was received for the given data set which had a mode of 1. The Client updates the mode to 2 and decrements the count of data sets that are still in mode 1 after displaying this message.

```
DataSet name[/rectype] will be cloned
```

This message, which applies to the `clone` command, indicates that the specified data set will be cloned. `/rectype` only appears for variable-format data set records that have a nonzero record type (contains a variable part).

```
DataSets initialized in DMSII
```

This message appears when the Databridge Engine sends the Client a status indication that one or more data sets have been initialized (emptied). This message is preceded by one or more of the following messages that occur during the purging of a data set: "Dropping *tablename*"; "Creating *tablename*"; and "Creating index *name* for *tablename*".

```
DB_Info: update_level = ddd, update_ts = timestamp, highest_strnum = ddd
DB_Info: database_ts = timestamp, database_name = name
[DB_Info: OptionFlags = options]
```



These messages are only written to the log file and only appear when one or more data sets is in tracking mode.

The first line provides the database update level, the database update timestamp, and the highest structure number in the database for the data source being accessed.

The second line gives the database timestamp and the database name for the data source being accessed.

The third line, which only appears when there is something to report, consists of a set of comma-separated option names. These include: `IndependentTrans` (`INDEPENDENTTRANS` is set for the DMSII database), `AccessActive` (`READ ACTIVE AUDIT` is set in the Engine Control File), `RDB` (the DMSII database is an RDB secondary database), `FileXtract` (the data source is a FileXtract file) and `LINKS` (`LINKS` is set to `TRUE` in the Engine Control File).

```
DB_Wait parameters: retry_secs = ddd, maxwait_secs = ddd, eee
```

This message, which is only written to the log file, shows the values of the `DB_Wait` RPC parameters when the configuration parameter `use_dbwait` is set to `True`. This information is logged as it one of the first thing we want to know, when the client appears to run very sluggishly. Setting `max_retry_secs` to a high value will make the client go idle for the specified amount of time when the Engine reaches the end of the audit trail.

When `eee` is 0 `ddd` is the time interval after which the Engine stops retrying when it finds no updates. On the other hand if `eee` is non-zero, it defines the time interval after which the Engine stops retrying when it finds no updates. In the latter case the client will complete the wait-and-retry loop by issuing `DBWait` RPCs until `ddd` seconds elapse with no updates received from the server. A value of 0 for `ddd` is taken to mean retry forever.

```
{DMAuditLib | FileXtract_Reader_Name} Version: version_string
```

This message, which is only written to the log file when the Client starts processing audit files, shows the reader being used for the audit files. When using `FileXtract`, the reader's name and version appear in the log file at the start of a `process` or `clone` command. These include `SUMLOG`, `TTRAIL`, `PRINTFIL`, `BICSS`, `DISKFILE`, `LINCLOG`.

```
DBServer TXN Group Statistics (cumulative):
```

```
Commits = ccc, Avg UPDATE_inc = uu.uu, Avg Trans time = ss.sss sec, Avg ABSN_inc = bb.bb
```

```
Rollbacks = rrr, Avg UPDATE_inc = uu.uu, Avg Trans time = ss.sss sec
```

This message displays cumulative statistics about commit and rollback operations performed under the direction of the Engine. `ccc` is total the number of commits; `rrr` is the total number of rollbacks; `uu.uu` is the average number of updates contained in the individual transactions; `ss.sss` is the average duration of these transactions; and `bb.bb` is average number of audit blocks the committed transactions span. These numbers are printed at the end of the incremental statistics.

```
Deleting file "name"
```

(Windows only) The specified file is being deleted. Instead of initiating the bulk loader, the temporary data files are deleted since the `-z` option implies that the database cannot be updated. This message only appears if you use the `-z` option.

```
DMSII item data: offset = dddd (0xhhh), len = dd half bytes  
0000 xx ...
```

This message, which is only written to the log file, appears after a data error warning if the parameter `display_bad_data` is enabled. These errors include bad digits in numeric data and control characters or 8-bit characters in ALPHA data. `xx` represent the DMSII values of the data bytes that make up this field.

Dropping table '*name*'

This message appears when a table gets initialized during the processing of updates, as a result of an INITIALIZE of the data set in DMSII. The client does this by dropping the table and recreating an empty table.

Effective COMMIT Parameters: BLOCKS = *bbb*, UPDATES = *uuu*, TRANS = *ttt*, ELAPSED = *eee*, LONG TRANS = {True | False}

This message, which is only written to the log file, shows the effective values of the various CHECKPOINT FREQUENCY parameters. The original values come from the Engine Control File, but are sometimes overridden by the values specified in the Client configuration file.

End fixup phase for cloned DataSets

This message indicates that the fixup phase for the cloned data sets is ending. At this point, all tables in the relational database that are mapped from active data sets are synchronized.

End populating/updating database at AFN=*afn*, ABSN=*absn*, SEG=*seg*, INX=*inx*, DMSII Time=*timestamp*

The `process` or `clone` command ends at the audit file location that corresponds to the given AFN, ABSN, SEG, INX, and DMSII timestamp values. The next `process` command starts at this point.

Extended translation library "*name*" successfully loaded and initialized

This message, which only applies when using an external translation DLL to perform the data translation, indicates the DLL was successfully loaded.

Filter file "*dbfilter.cfg*" successfully processed

This message, which is written to the trace file when the TR\_VERBOSE bit in the trace mask is set, indicates that the binary was successfully loaded and the associated data structures were successfully initialized.

Incremental Statistics (AFN=*n*):

This message contains a set of incremental statistics, which apply only to each individual audit file processed. The AFN displayed identifies the audit file for these statistics. It is the previous audit file, as we print these statistics when we encounter the first quiet point in an audit file.

Processed *nnn.nn* KB in *sss.sss* secs, throughput = *ddd.dd* KB/sec, DMSII recs/sec = *rrr.rr*, lag time = *hh:mm:ss*

*nnn.nn* represents the number of kilobytes of DMSII data received. *sss.sss* represents the elapsed time in seconds. *ddd.dd* represents the corresponding throughput. The lag time, which is represented as hours, minutes and seconds, is the difference between the times when the record is updated in the relational database and when it was updated in DMSII (this is only meaningful when doing real/time replication).

Received *nnn.nn* KB from DBServer in *sss.sss* secs, total throughput = *ddd.dd* KB/sec

*nnn.nn* represents the number of kilobytes of total data received (including the packet headers and the non-data packets). *sss.sss* represents the elapsed time in seconds. *ddd.dd* represents the corresponding throughput. *rrr.rr* represents the corresponding rate at which DMSII records were processed.

DMSII Buffers Used = *dd* (configured\_max = *mm*), Audit access rpc = {DBRead | DBWait}, Audit file origin = *AF\_origin*

*dd* is the actual number of DMSII buffers used and *mm* is the configured maximum. If the configured value is 0, the maximum value is computed by the program. The rest of the line indicates whether DBRead or DBWait was used to get updates from the Databridge Engine and the method by which the audit file is being read. The possible values for *AF\_origin* are HostAudit, DirectDisk, IndirectDisk, and DBECache (the last three apply to Enterprise Server).

TCP/IP time = *sss.sss* secs, (*dd.dd*% of total time)

*sss.sss* represents the amount of time (in seconds) that the program spent waiting for TCP/IP data to appear from the host. *dd.dd* is the corresponding percentage of total elapsed time.

SQL\_exec time = *sss.sss* secs, (*dd.dd*% of total time)

*sss.sss* is the amount of time (in seconds) that the main thread of the program spent waiting for the execution SQL statements to complete. *dd.dd* is the corresponding percentage of total elapsed time.

TXN\_exec time = *sss.sss* secs, (*dd.dd*% of total time)

*sss.sss* represents the amount of time (in seconds) the main thread of the program spent waiting for commits to complete. *dd.dd* is the corresponding percentage of total elapsed time.

[buf\_wait time = *sss.sss* secs, (*dd.dd* % of total time)]

This line only applies to multi-threaded updates. *sss.sss* represents the amount of time (in seconds) the main thread of the program spent waiting for a DMSII Buffer to become available. *dd.dd* is the corresponding percentage of total elapsed time. DMSII buffers are used to hold the raw DMSII records while they are being processed by the client. In the case of multi-threaded updates these buffers are placed on the various updater threads' work queues using working storage header blocks described below.

[ws\_wait time = *sss.sss* secs, (*dd.dd* % of total time)]

This line only applies to multi-threaded updates. *sss.sss* represents the amount of time (in seconds) that the main thread of the program spent waiting for a working storage header block to become available. *dd.dd* is the corresponding percentage of total elapsed time. Working storage header blocks are small blocks used to queue DMSII buffers, as they can be on multiple queues when a DMSII data set maps to multiple tables.

[thr\_wait time = *sss.sss* secs, (*dd.dd* % of total time)]

This line only applies to multi-threaded updates. *sss.sss* represents the amount of time in seconds that the main thread of the program spent waiting for the updater threads to finish processing updates.

CPU/other time = *sss.sss* secs, (*dd.dd*% of total time)

*sss.sss* represents the amount of time (in seconds) the main thread of the program was not waiting for any of the above actions to complete. *dd.dd* is the corresponding percentage of total elapsed time. This metric is a derived value and it represents the time during which the thread was in a runnable state.

```
Server_Packet_Counts:  
CREATE:ddd   DELETE:ddd   MODIFY:ddd   MOD_BI:ddd   MOD_AI:ddd  
STATE:ddd  
LINK_BI:ddd  LINK_AI:ddd   BULKDEL:ddd  DOC:ddd    COMMIT:ddd  
ROLLBK:ddd
```

These values represent the counts of the various record types and commits received from DBServer. CREATE represents insertions into the database. DELETE and MODIFY represent delete and update operations during audit file processing. MOD\_BI and MOD\_AI represent updates that are before and after image records for updates. BULKDEL will normally be 0, as it is only used when implementing embedded subsets. STATE represents StateInfo records. LINK\_BI and LINK\_AI represents before and after images of link items. COMMIT represents commits. ROLLBK represents rollbacks. DOC represents documentation records mostly used for debugging. The two exceptions are DOC records that provide information on an audit file, when the Engine starts reading a new audit file and DOC records that indicate that a data set has been reorganized (this is used to inform the client that a garbage collection has occurred, as it otherwise would not know about it).

```
Server_Rolledback_Packet_Counts:  
CREATE:ddd   DELETE:ddd   MODIFY:ddd   MOD_BI:ddd   MOD_AI:ddd   STATE:ddd  
LINK_BI:ddd  LINK_AI:ddd   BULKDEL:ddd  DOC:ddd
```

These values are printed when rollbacks occur and represent the counts of the record types that were rolled back.

```
[Processed: dddd bytes of before image data, rrrr redundant SQL updates  
skipped]
```

This line only appears when using the `optimize_updates` feature. *dddd* represents the number of bytes of DMSII before image data received (the cause) and *rrrr* represents the number of redundant SQL updates that were eliminated (the effect). A low value of *rrrr* and a high value of *dddd* is a clear indication that the `optimize_updates` feature is not helpful in this case.

```
[Data Errors: eeeeee SQL rows discarded, dddddd SQL rows in error]
```

This line shows the number of records that had data errors or were discarded. It doesn't appear when both counts are zero.

```
[Filter: dddddd occurs tables records suppressed]
```

This line shows the number of records that were discarded from OCCURS tables because of filtering. It doesn't appear when the count is zero.

```
Index 'name' for table 'name' created successfully
```

This message is printed when the index creation is successful.

```
Initiating process command for DataSource name
```

This message indicates that a scheduled `dutility process` command (via the scheduling parameters in the `dbridge.cfg` file) is about to begin.

```
Key change detected in MODIFY for DataSet name[/rectype], handling it as a DELETE/  
INSERT instead
```

This message, which only appears in the trace file when the TR\_VERBOSE bit in the trace mask is set, indicates that the Client detected a change in the value of one or more `key_items`. At the start of a `process` or `clone` command, the Databridge Client registers the keys being used with the Databridge Engine for data sets that use a SET that has the KEYCHANGEOK attribute as the source for the index. The Databridge Engine handles this situation by sending the updates to the client as MODIFY records when the keys are unchanged or as MODIFY BI/AI pairs when the keys change. For information about the DSOPT\_CheckKeyChanges bit in "ds\_options" and "check\_key\_changes", refer to the *Databridge Client Administrator's Guide*.

Launching refresh command to update stored procedures

This message indicates that dbfixup set a bit in the data source's `status_bits` column to indicate that there are OCCURS tables present. Upon seeing this bit a `process` or `clone` command initiates a `refresh` command to get the stored procedures `z_tablename` created. These stored procedures are used to speed up delete operations for such tables. Rather than deleting the rows of secondary table for a given key one by one, we delete them all in a single SQL statement using this stored procedure.

Loading binary filter file "dbfilter.cfg"

This message, which will only be displayed when there is an occurs table filter present in the config subdirectory for the data source. The filter file is always named "dbfilter.cfg". This message is a simple confirmation that the binary filter file was read by the client. Refer to the section on OCCURS table filtering in the Client Administrator's guide for detail on how this type of row filtering works.

Loading control tables for *datasource*

This message indicates that the Client control tables are being loaded for the data source you specified with the `process` or `clone` command.

Log file switched to "filename" (*reason*)

This message appears under the following conditions, which cause the Client to close the current log file and open a new one:

The `logsw_on_size` configuration parameter is set to True and file size exceeds the configured maximum during an audit file switch.

The `logsw_on_newday` configuration parameter is set to True and the Client notices that the date has changed.

Mainframe Time '*hh:mi:ss*'; {ahead | behind} by *hh:mi:ss*

This message, which appears at the start of a process command, shows the time difference between the mainframe and the Client machine clocks. This value is factored in to all lag time calculations. The Client periodically checks the mainframe clock, to prevent the lag time from going negative if the clocks are adjusted. You will therefore see this message multiple times during the course of a long Client run.

We always display this message at the start of the run and from thereon only if it drifts by more than 2 seconds.

MODIFY occurs depending on, item = 'name', bi\_count = *ddd*, ai\_count = *nnn*  
- Keys: *columnname = value,...*

This message, which only appears in the trace file when the TR\_VERBOSE bit in the trace mask is set, shows the old and new values of the depends item for an OCCURS DEPENDING ON clause. The values determine how the update is handled when the OCCURS is not flattened.

Next update for DataSource *name* will run at *hh:mm* (delay = *nn* secs)

This message appears only when you have scheduling parameters enabled for `dbutility` in the configuration file. It tells you when to expect the `process` command to run again. *hh:mm* corresponds to the time at which the next run starts and *nn* represents the length of this delay (in seconds).

Processed: *nnnn* DMS recs, *rrrr* SQL rows

This message appears if the `display_statistics` configuration parameter or the `-v` option is enabled. This message is useful when you are not sure if the client is running or if it has stopped operating, especially when the cloning requires several hours.

Processed: *nnnn* DMS recs,*rrrr* SQL rows[, *bbbb* SQL rows rolled back]

This message, which is printed after the cumulative statistics, contains the total count of DMSII records that were processed and the corresponding count of SQL updates. In addition, when rollbacks occur, it displays the count of rows rolled back.

Processing updates from: AFN=*afn*, ABSN=*absn*, SEG=*seg*, INX=*inx*, DMSII  
Time=*timestamp*

This message appears after the incremental statistics, which are displayed when the client encounters the first quiet point in a new audit file. This message indicates that the Client is processing updates from the specified AFN, ABSN, SEG, INX, and DMSII time stamp. The statistics that precede it apply to the previous audit file.

QUIT command initiated by QUIT signal

This message, which is limited to the UNIX client, indicates that a kill command was used to generate a SIGQUIT signal, which the client is responding to. The Client treats this signal exactly like a command line console QUIT command. This is particularly useful if you are running the Client as a background run.

Redundant update for table '*name*'; Keys: *columnname* = *value*,...

This message only appears in the trace file when the following conditions are met: the `TR_VERBOSE` bit in the trace mask is set; the configuration file parameter `optimize_updates` is set to `True`; and the data set has been marked to receive before- and after-images. The message indicates that the Client found no value changes in the columns of the table and no update is needed. This means that the update can be skipped as it does nothing.

ReleaseSemaphore for *name* reached maximum value, retrying after brief delay

(Windows only) This message, which only appears in the trace file when the `TR_VERBOSE` bit in the trace mask is set, indicates that the Client has exceeded the maximum posting limit for the semaphore by attempting to post too many requests. The Client will stall until the thread that handle requests catches up with the requests. In the case of `bcp_work_semaphore`, this usually means that you have attempted to clone a large number of empty data sets. The client should normally recover from this situation.

Rerunning the script to clear duplicate records for table '*name*'

(SQL Server only) This message indicates that the script which clears duplicate records got a schema change error (539); that is, the schema of tables changed while a select into statement was executing. The Client recovers from this error by rerunning the script. This error only occurs on high-end multiple-CPU machines. If the script fails a second time, the Client returns the message, "WARNING Attempt to clear duplicate records for table '*name*' failed." In this case, the `ds_mode` for the data set will be set to 11 and the data set will not be tracked.

Restarting fixup phase for previously cloned DataSets

This message appears at the start of a `process` command (after the message "Begin populating/updating database from AFN = afn, ABSN = absn, ...") if there are data sets in fixup mode (`ds_mode = 1`) and no data sets need to be cloned (`ds_mode = 0`).

Restarting process command for data source *name*

This message, which is limited to the command line client `dbutility`, indicates that the Databridge Engine returned an error status that is handled by restarting the Client after a brief delay. These errors all have to do with race conditions when the Engine is attempting to read the current audit file before DMSII has finished writing it. When using the Client Manager Service you will never see this message, as the service handles the error recovery.

Rows loaded by table:

<i>name</i>	<i>rowcount</i>	<i>name</i>	<i>rowcount</i>	<i>name</i>	<i>rowcount</i>
-------------	-----------------	-------------	-----------------	-------------	-----------------

This message, which is only written to the log file at the end of the data extraction phase when the `show_table_stats` parameter is set to True, shows the number of rows loaded for the each of the affected tables.

Running script "*script\_file\_spec*"

This message appears only when a data table creation user script or an index creation user script is run.

Selecting DataSet *name*[/rectype]

This message, which only appears in the trace file when the `TR_VERBOSE` bit in the trace mask is set, specifies that the data set in question is being selected at the beginning of a `process` or `clone` command. It gives you a chance to make sure that you have selected all of the data sets you want.

SQLCancel operation completed

The Client's timer thread monitors SQL queries that fail to complete within a reasonable amount of time controlled by the configuration parameter `sql_exec_timeout`, which contains two values.

The first value is the time interval after which the timer issues a warning about the SQL operation taking longer than anticipated. The second value is the time after interval after which the timer thread initiates steps to remedy the situation. The first step is to issue a database API call to cancel the query. In the case of SQL Server this is the ODBC `SQL_Cancel` procedure.

We have observed that in a situation where the client is run in a virtual machine (VM), this operation sometimes also hangs as the problem is due to a loss of connectivity between the ODBC driver and the database machine due to a problem with the VMWARE networking that occasionally gets into this situation. This message indicates that the `SQL_Cancel` call did not hang. In order to prevent the timer thread from hanging the SQL Server client creates a temporary thread to issue the `SQL_Cancel` call. The timer thread then monitors this thread and if it detects that the `SQL_Cancel` is hung it kills the Client task allowing the service (or the script file that launched the client) to detect the situation and restart the client, which normally resolves the issue.

Starting bcp for table '*name*'

(SQL Server Client only) This message indicates that the Client opened a BCP API connection and is about to start loading the specified table.

Starting { `bcp | sql*loader` } for table '*name*'

(Windows only) This message applies when you are using the /z option. It is only displayed in the trace file when the TR\_VERBOSE bit in the trace mask is set. It indicates that the Client is starting the bulk loader for the given table.

Starting command: "*filename*"

(UNIX only) The given shell script that runs the bulk loader is being started. The shell script (*load.tablename.sh*) runs the bulk loader and establishes a named pipe used to communicate data between the main process and the bulk loader process that is spawned from the script.

Starting fixup phase for cloned DataSets

This message indicates that the Client has finished extracting data and is starting the fixup phase for the cloned data sets.

Stopping: All available audit information has been processed (*status*)

This message indicates that no more audit file information is available on the host (that is, a normal stopping point). When READ\_ACTIVE\_AUDIT is set to FALSE in the Engine Control file, or when audit files are available but there are no updates, the Databridge Engine will also cause the Client to display this message and stop.

**NOTE:** If the Databridge Engine finds updates but reaches the end of the audit file before a commit, the Client rolls back the updates. The discarded updates are included when the next audit file or quiet point becomes available.

The status received from DBServer is one of the following:

AUD\_UNAVAIL indicates a normal exit.

LIMIT\_NAME indicates that the run stopped because the Databridge Engine encountered a task name that satisfied the stop condition. This condition can be specified using a STOP statement in the DBServer parameter file or the Client configuration file.

LIMIT\_TIME indicates that the run stopped because the Databridge Engine is processing an audit file record whose time stamp satisfies the stop condition. This condition can be specified using a STOP statement in the DBServer parameter file or the Client configuration file.

Stopping: Audit information not available (*status*)

For an explanation of statuses, see the preceding message. This message occurs when no audit files on the host have been read by the Databridge Engine. This can indicate that no audit files are available (that is, no audit file is closed) or that the Databridge Engine does not have visibility to the audit files. In this case, try again when an audit file is closed. If READ\_ACTIVE\_audit is set to FALSE in the Engine Control file, you will get this message when the client tries to open the active audit.

Stopping: Client operations inhibited between *hh:mm* and *hh:mm*

This message indicates that the Client stopped the processing of audit files because it is entering a blackout period defined by the configuration parameter `blackout_period` in the scheduling section.

In the case of dbutility, this message also applies to a blackout period defined using the `stop_time` and `end_stop_time` columns in the DATASOURCES table entry for the data source. To use this feature, which can be associated with the shutdown parameter, set the configuration parameter `controlled_execution` to True.



Stopping: Database reorganization - execute a redefine command followed by a reorg command

This message appears when you run a process command, and the Databridge Engine detects a DMSII structural reorganization of filler substitution that occurred. See "Changes to the DMSII Database" in Chapter 4 of the *Databridge Client Administrator's Guide*.

Stopping: Database update level changed -- execute a redefine command

This message appears when you run a process command and the Databridge Engine detects a DMSII structural reorganization that does not affect any of the selected data sets. In this situation, the client will stop, except when you use a MISER database, in which case it will display the warning, "Database update level change ignored".

When the Client stops after a database update level change, it forces you to run a redefine command. This ensures that DASDL changes that affect the operations of the Databridge Client are addressed, even if those changes don't result in a format level change for the data set.

For example, if an item is added to the SET which is used by the Client as the source of the index, that item would not be discovered by the Client. Instead, the Client would create false duplicate records which would result in missing records in the relational database. Another example is if changes are made to the KEYCHANGEOK attribute of the SET that Client uses as the source for index. The Client would not know that KEYCHANGEOK had been set and would create extra records in the relational database when a key change actually occurs.

Stopping: DBEnterprise audit file origin changed

This message appears when you run a process command and the Databridge Client detects a change in the audit file origin (access method) when using Databridge Enterprise Server. When starting to process a new audit file the client gets a DOC record that has information about the audit file. This includes the access method (direct disk, indirect disk or cache), which defines how the audit file is being read. The configuration parameter stop\_on\_dbe\_mode\_chg determines whether or not the client should stop when it detects that the audit file origin no longer matches the value specified in the configuration parameter dbe\_dflt\_origin.

Stopping: Discard threshold exceeded

This message indicates that the client has exceeded the limit on the total number of discard records specified in the configuration file using the first value of the parameter max\_discards. This situation results in exit code 2054.

Stopping: Errors occurred during data extraction [and index\_creation]

This message appears before the fixup phase if discard files were created and (if the last part of the message is present) index creation errors occurred. Instead of entering the fixup phase, the program stops and gives you a chance to look into the problem before continuing.

Stopping: Errors occurred during index creation

This message appears before the fixup phase if index creation errors occurred. Instead of entering the fixup phase, the program stops and gives you a chance to look into the problem before continuing.

Stopping: Garbage collection reorganization has occurred

This message indicates that the processing of updates is being interrupted. This message appears at the first quiet point after a garbage collection reorganization if the stop\_after\_gc\_reorg parameter is enabled.

Stopping: Operator issued a "quit" command

This message indicates that a `QUIT` command was issued for the Client.

Stopping: Processing of `fixup` records deferred to next process command

This message appears when the program typically enters the `fixup` phase if the configuration parameter `defer_fixup_phase` is enabled (the `-c` option toggles this command). Instead of entering the `fixup` phase, the program stops. This procedure is useful when using parallel cloning.

Stopping: Processing of updates deferred to next process command

This message, which appears at the end of the `fixup` phase if the `stop_after_fixups` parameter is enabled, indicates that updates will be processed at the next `process` command.

Stopping: Processing through requested AFN completed

This message appears when the value of an audit file goes past the audit file stored in the `stop_afn` column of the `DATASOURCES` Client control table. It only appears if the configuration parameter `stop_after_given_afn` is enabled. The program resets this column to zero before stopping, which removes the stopping condition.

Temporary storage threshold reached, starting bulk loader

(Windows only) This message indicates that the client has reached the bulk loader `max_temp_storage` cutoff, which is half the value of this parameter. All tables for which temporary files were created will be queued for loading and the main thread of the client will continue processing extracts until the full threshold is reached. At that point, it will block waiting for the bulk loader thread to finish loading the tables. If the bulk loader thread finishes before this happens, the client looks for cutoff at which point it repeats the aforementioned process.

Thread[*nn*]:

SQL\_time = *sss* secs, (*nn.nn*% of total time), update\_cnt = *ddd*, average  
update\_time = *m.mmm* ms

[BCP\_time = *sss* secs, (*nn.nn*% of total time), record\_cnt = *ddd*, average load\_time  
= *m.mmm* ms]

[FileIO\_time = *sss* secs, (*nn.nn*% of total time), record\_cnt = *ddd*,  
average IO\_time = *m.mmm* ms]

Idle\_time = *sss* secs, (*nn.nn*% of total time)

Other\_time = *sss* secs, (*nn.nn*% of total time)

This message only appears when multi-threaded updates are enabled. It is appended to the data extraction statistics and is included in the incremental and cumulative statistics (after `thr_wait_time`). There is one message for each update thread, where *nn* is the thread number (1 to 16).

In the data extraction statistics the line `SQL_time` will be all zeroes, unless you loaded some tables without using the bulk-loader. The value *sss.sss* is the amount of time (in seconds) that the thread spent waiting for the execution SQL statements to complete. *dd.dd* is the corresponding percentage of total elapsed time. The `update_cnt` parameter represents the number of updates that the thread executed since the start of the last audit file and `average_update_time` represents the average duration of the updates expressed in milliseconds with three fractional digits.

In the line `BCP_time`, *sss.sss* is the amount of time (in seconds) that the thread spent waiting for the BCP API to load records. *dd.dd* is the corresponding percentage of total elapsed time. It is followed by the `record_cnt`, which represents the number of records that the thread loaded and `average_load_time` represents the average duration of the load operations expressed in milliseconds with three fractional digits.

In the line `FileIO_time`, *sss.sss* is the amount of time (in seconds) that the thread spent waiting for the file I/O operations to complete. *dd.dd* is the corresponding percentage of total elapsed time. It is followed by the `record_cnt`, which represents the number of records that the thread loaded and `average_load_time` represents the average duration of the load operations expressed in milliseconds with three fractional digits.

In the line `Idle_time`, *sss.sss* represents the amount of time (in seconds) that the thread spent waiting for work. *dd.dd* is the corresponding percentage of total elapsed time.

In the line `Other_time`, *sss.sss* is the amount of time (in seconds) that the thread was not waiting for any of the above actions to complete, and *dd.dd* is the corresponding percentage of total elapsed time. This metric is a derived value and it represents the time during which the thread was in a runnable state.

```
Total thread BCP_time = sss secs, total extract_cnt = dddd, average sql_time =  
m.mmm ms
```

This message appears at the end of the thread statistics that are printed at the end of data extraction when using multi-threaded extracts (i.e. `n_update_threads` is > 0). They represent the statistic across all threads.

```
Total thread SQL_time = sss secs, total extract_cnt = dddd, average sql_time =  
m.mmm ms
```

This message appears at the end of the thread statistics that are printed at the end of data extraction when using multi-threaded extracts (i.e. `n_update_threads` is > 0). They represent the statistic across all threads for load operations that do not use the bulk-loader or the BCP API in the case of the SQL Server Client.

This message also appears at the end of the thread statistics that are printed at the end of the run or after the Client starts processing a new audit file when using multi-threaded updates (i.e. `n_update_threads` is > 0). They represent the statistic across all threads.

```
TranCommit, AFN=afn, ABSN=absn, SEG=seg, INX=inx, DMSII Time=timestamp
```

This message, which only appears in the trace file when the `TR_VERBOSE` bit in the trace mask is set, indicates that the last group of SQL statements were committed on the relational database, along with the DMSII time and the ending audit file stateinfo.

```
Transaction group aborted by DBEngine, nnn operations rolled back
```

This message appears when the Databridge Engine sends the Client an abort status (instead of a commit status). The Databridge Engine rolls back the current transaction group. Transaction rollbacks can happen when the Engine reaches the end of the audit trail, or when `READ_ACTIVE_AUDIT` is set to `FALSE` in the Engine Control file and the Engine reaches the active audit file. In the cases, these updates are applied the next time the client is run. Additionally, programs on the mainframe that roll back updates can sometimes cause the Engine to send the Client rollback requests. When rollbacks occur, they are included in the statistics that the Client logs. See “Incremental Statistics” in this section.

Updates by table (average update times in ms):

```
name      updates (m.mmm)   name      updates (m.mmm)   name      updates (m.mmm)
```

This message, which is only written to the log file after an audit file switch when the `show_table_stats` parameter is set to `True`, shows the number of updates and the average update time for each table during the processing of the last audit file in question. Tables that have no updates are omitted from this list.

## Drop and Dropall Commands Messages

The following messages appear in response to the Databridge Client `drop` or `dropall` commands.

You must create a separate working directory for each data source. When you need to drop a data source, make sure that the current directory is the working directory for the data source before running the `drop` command. This ensures that `dbutility` will be able to locate the required scripts. In the case of a `dropall` command, you can only drop one data source, as the working directory must be changed for each data source. This command will also drop the Client control tables when dropping the last data source. The `dropall` command is rarely used in this way. It is mainly used for dropping the Client control tables.

Cleaning up table '*name*'

This message indicates that the Client is deleting selective records from the specified table instead of dropping the table. This action is only taken in special cases, such as when tables that contain non-DMSII data are populated by the Client. The `drop` command cannot drop the table; instead, it removes all the records that the Client created.

Deleting control table entries for DataSource *name*

This message indicates that all of the Client control table entries related to the specified data source are being removed.

Drop of all Databridge tables successfully completed

This message indicates the following:

- ◆ All tables for the various data sources defined in the Client control tables have been removed from the relational database.
- ◆ All of the corresponding scripts have been deleted from the current directory.
- ◆ All of the Client control tables have been removed from the relational database.

Drop of DataSource *name* partially completed

The `drop/dropall` command deletes all tables, stored procedures, and scripts for each table.

If no error occurs, the `drop/dropall` command proceeds to delete the Client control table entries corresponding to the specified data source.

If an error occurs while deleting the Client control table entries, the `drop/dropall` command continues trying to delete the Client control table entries for the data source from the remaining tables. In this case, you must manually remove the remaining tables, stored procedures, and/or scripts, as well remove the corresponding entries in the Client control tables.

Drop of DataSource *name* successfully completed

This message indicates that the tables and associated stored procedures for this data source have been removed, as well as the data source entry in the Client control tables. In addition, scripts for this data source were deleted from the scripts subdirectory.

Dropping control table *name*

This message is used by the `dropall` command to indicate that the Client control table in question is being removed from the relational database in the final stages of the command.

Dropping table '*name*'

This message, which only appears in the trace file when the `TR_VERBOSE` bit in the trace mask is set, indicates that the specified table and its associated stored procedures are being removed from the relational database. This message is also used by the `dropall` command to indicate that the specified Client control table is being removed from the relational database in the final stages of the command.

Loading control tables for *datasource*

This message indicates that Client control tables are being loaded for the data source that you specified (with the `drop` or `dropall` command).

Starting drop of DataSource *name*

This message indicates that the `drop` (`dropall`) command has begun for the specified data source.

## Switchaudit Command Messages

The following message may appear in response to the Databridge Client `switchaudit` command.

Audit file switched (Current DMSII AFN = *nnnn*)

This message indicates that the DMSII audit file was closed and a new one was opened. If the `READ ACTIVE AUDIT` parameter is set to `FALSE` in the Engine Control file, the Databridge Engine will not attempt to process the active audit file, unless you are doing a clone. The value *nnnn* is the DMSII database's current audit file number after the command completes.

## Display Command Messages

The following messages appear in response to the Databridge Client `display` command.

Control tables for DataSource *name* written to file "*name*"

This is the confirmation message indicating the successful completion of the command. This message is output to the screen, not the Client log file.

Loading control tables for *datasource*

This message indicates that the Client control tables are being loaded for the data source you specified with the `display` command.

## Runscript Command Messages

The following messages appear in response to the Databridge Client `runscript` command.

Running script "*script\_file\_spec*"

This message is displayed when the Client runs the specified script.

Script SQL statements and row counts will be written to file "*name*"

This message is a reminder that the `runscript` command automatically enables SQL tracing and also writes the row counts for insert, delete, and update SQL statements executed in the script. The row counts are of the form "*nn* row(s) {inserted | deleted | updated}".

User script "*name*" executed successfully

This message indicates that the specified script ran with no errors.

## Unload Command Messages

The following messages appear in response to the Databridge Client `unload` command.

Control tables for all DataSources written to file "*name*"

This message is displayed at the end of an `unload` command when the data source name field of the command line contains the value `_all`. This message indicates that the command completed successfully.

Control tables for DataSource *name* written to file "*name*"

This message can arise in two situations:

If a data source name is specified on the command line, this message indicates that `unload` command completed successfully.

—or—

If the data source name field of the command line contains the value `_all` and the `d` option is used, this message is displayed after each data source is unloaded.

Loading control tables for *datasource*

This message is displayed when the program loads the Client control tables for the specified data source before writing their records out to a file. If several data sources are being unloaded, this message displays multiple times.

Unloading control tables for *datasource*

This message indicates that the `unload` command is writing the control tables entries for the given data source to the file in question.

## Reload Command Messages

The following messages appear in response to the Databridge Client `reload` command.

Control tables for all DataSources reloaded from file "*name*"

This message will appear at the end of a `reload` command if the data source name field of the command line contains the `_all` value. It indicates that the command completed successfully.

Control tables for DataSource *name* reloaded from file "*name*"

This message appears at the end of a `reload` command if a data source is specified on the command line. It indicates that the command completed successfully.

DataSet *name*[/*rectype*] will be reloaded

This message appears if a data set list is specified in the command line of a `reload` command. It is a confirmation message printed prior to reloading the Client control table entries that pertain to the data set in question.

Loading control tables for *datasource*

This message is displayed if a data set list or the `-k` option is specified on the command line of the `reload` command. The command needs to first load the Client control tables to determine if the specified data sets exist and to possibly preserve the state information.

Reloading Control table entries for DataSource *name* from file "*name*"

This message can appear in two situations:

If a data source name is specified on the command line, this message indicates that `reload` command is about to reload the control tables for the data source in question.

—or—

If the data source name field of the command line contains the value `_all`, this message is displayed before each data source is reloaded. There is no confirmation message in this case, except at the very end of the command.

## Refresh Command Messages

The following messages appear in response to the Databridge Client `refresh` command. This command is normally embedded in the `reorg` command. If you decide to manually process a DMSII reorganization by writing your own `alter table` commands, you will need to run this command. Make sure that you first execute a `generate` command to ensure that the Client scripts are up-to-date.

Loading control tables for datasource

This message indicates that client control tables are being loaded in preparation of the execution of the `refresh` command.

Script SQL statements executed will be written to file "*name*"

This message is a reminder that the `refresh` command automatically enables SQL tracing when the `-v` option is enabled. The `refresh` command drops the stored procedures for all the tables mapped from the specified data set and then recreates them. If a variable-format data set is specified, all data sets with the given name that have their active column set to 1 in the DATASETS Client control table are refreshed (the stored procedures of all replicated record types are refreshed).

Stored procedures for all active tables of DataSource *name* successfully refreshed

This message indicates that the `refresh` command, which has a data set name specification of "`_all`", completed successfully. It confirms that all stored procedures for all active tables in the specified data source were successfully refreshed.

Stored procedures for all tables of DataSet *name*[/*rectype*]successfully refreshed

This message confirms that the `refresh` command completed successfully for the data set that you specified on the command line.

## Export Command Messages

The following message may appear in response to the Databridge Client `export` command.

Text configuration file "*name*" successfully created

This message is confirmation that command completed successfully. If using the defaults, the binary file `dbridge.cfg` is read and its equivalent text configuration file `dbridge.ini` is created in the config subdirectory for the data source.

## Import Command Messages

The following messages appear in response to the Databridge Client `import` command.

Binary configuration file "*name*" successfully created

This message confirms that the command completed successfully. If using the defaults, the text configuration file "`dbridge.ini`" is read and its equivalent binary file "`dbridge.cfg`" is created in the config subdirectory for the data source.

## Rowcounts Command Messages

The following message may appear in response to the Databridge Client `rowcounts` command.



Loading control tables for *datasource*

This message indicates that client control tables are being loaded in preparation of the execution of the `rowcounts` command.

Row counts for all active tables for DataSource *name* written to file "*name*"

This message confirms that the command completed successfully. This message confirms that the command completed successfully. The row counts of all user tables, whose active column is 1 in DATATABLES, that are associated with data sets that have their active columns set to 1 in DATASETS are written to the client log file.

Table row counts:

<i>name</i>	<i>rowcount</i>	<i>name</i>	<i>rowcount</i>	<i>name</i>	<i>rowcount</i>
.	.	.	.	.	.

This message consists of the names of the various tables followed by their row counts. There are 3 entries per line.

## CreateScripts Command Messages

The following message may appear in response to the Databridge Client `createscripts` command.

Backing up user scripts to directory "*name*"

This message indicates that the old user scripts are being copied to the specified directory. Unless inhibited by the command line `-n` option, user scripts are backed up to the "*datasourceYYYYMMDD[\_HHMISS]*" subdirectory of the directory whose name is specified by the configuration parameter `user_script_bu_dir`. If this parameter is not specified, the directory specified by the parameter `user_script_dir` is used instead.

**NOTE:** You need to periodically delete old copies of these directories as the client does not try to manage the backup user scripts directories.

Creating DataSet selection script

This message indicates that the program is creating the script "`script.user_datasets.source`". This script normally contains data set selection scripts which manipulate the active column in the DATASETS control table.

Creating user define script for DataSet *name*[/*rectype*]

This message indicates that the program is creating the script "`script.user_define.dataset`", where *dataset* is the name of the primary table for the specified data set. These scripts typically perform renames of tables and columns. This is achieved by updating the DATATABLES and DATAITEMS control tables.

Creating user layout alteration script for DataSet *name*[/*rectype*]

This message indicates that the program is creating the script "`script.user_layout.dataset`", where *dataset* is the name of the primary table for the specified data set. These scripts typically modify the `di_options` and `dms_subtype` columns in the DMS\_ITEMS control table to perform customizations, such as cloning numbers as dates or flattening OCCURS clauses.

Loading control tables for *datasource*

This message indicates that client control tables are being loaded for the data source you specified with the `createscripts` command.

User scripts for DataSource *name* written into directory "*name*"

This message confirms that a fresh copy of the user scripts were written into the specified directory. The client removes all user scripts from the directory after backing them up and recreates all the user scripts.

**NOTE:** If you set active to 0 for a data set prior to running the `createscripts` command, the user scripts for that data set will not be created. You might need to retrieve them from the backup directory if you later change the active column back to 1. Alternatively, you can run the command again with the `-n` option to prevent a second backup copy from being created.

## Tcptest Command Messages

The following messages appear in response to the Databridge Client `tcptest` command.

```
Bytes Processed nnn.nn KB of DMSII data in sss.sss secs, throughput = ddd.dd KB/sec
Bytes Received nnn.nn KB in sss.sss secs, total throughput = ddd.dd KB/sec
TCP/IP_time = sss.sss secs, (dd.dd of total time)
```

This message appears at the end of the `tcptest` command.

In the first line, *nnn.nn* represents the number of kilobytes of simulated DMSII data received. *sss.sss* represents the elapsed time (in seconds). *ddd.dd* represents the corresponding throughput.

In the second line, *nnn.nn* represents the number of kilobytes of data received. *sss.sss* represents the elapsed time (in seconds). *ddd.dd* represents the corresponding throughput. The number of bytes received is slightly greater than the number of bytes of simulated DMSII data as it also includes the protocol overhead bytes.

In the third line, *sss.sss* is the amount of time (in seconds) that the program spent waiting for TCP/IP data to appear from the host. *dd.dd* is the corresponding percentage of total elapsed time.

```
TCP Test completed successfully
```

This message indicates that the Client has successfully completed the `tcptest` command.

```
TCP_Test: len=nnnn, count=nnnn
```

This message indicates the length of each message and the number of messages sent.

```
TCP_Test: nnnn iterations completed
```

The Client displays this message while executing the `tcptest` command if the `show_statistics` parameter is set to True. The Client displays this message after every *nnnn* iterations, where *nnnn* is the smallest value specified for the `statistics_increment` parameter argument.

## Databridge Client Console Messages

The following messages appear in response to `dbutility` console commands and command issued from the Client Console.

```
Aux STMT Statistics:
Configured_max = nnn, Max_used = mmm, Recycled_stmt_cnt = rrr.rr
STMT reuse stats: min_sql_ops = nnn, max_sql_ops = mmm, avg_sql_ops = rrr.rr
STMTs never reused = nnn, min_sql_ops = mmm, max_sql_ops = ddd, avg_sql_ops =
rrr.rr
```

This message appears in response to a `dbutility ASTATS` console command. For details, see [“Process and Clone Commands Messages” on page 66](#).

```
Client State = state_name
```

This message shows the Client status during a `process` command. It is displayed by `dbutility` in response to a `STATUS` command and by the DBClient in response to a `Query Client Status` command from the Client Console. This message typically includes the following additional lines. (The lines "Waiting for ..." and "Thread[n]" are described elsewhere in this section.)

```
Processing updates from AFN=afn, ABSN=absn, SEG=seg, INX=inx, DMSII
Time=timestamp [(lag time = hh:mi:ss)]
```

These lines display the current `StateInfo`, and when processing audit files, is followed by the lag time.

```
Server = {DBServer | DBEnterprise}, Audit_access_rpc = {DBRead | DBWait
(retry_secs = mmm, maxwait_secs = nnn, nnn)} [, Audit file origin =
AF_origin]
```

These lines provide information about the RPC being used to read the audit file (in the case of `DBWait` we also display the parameters of the RPC) and the origin of the audit file updates. Possible values for `AF_origin` include `DBServer`-based operations `HostAudit` and `Databridge Enterprise Server`-based operations `IndirectDisk`, `DirectDisk` and `DBECache`.

```
RPC Info: last_rpc = name, send_seq_no = 0xhhhh, recv_seq_no = 0xhhhh,
resp_count = dddd
```

This line is included to help in diagnosing occasional issues with the communication between the Client and Enterprise Server hanging with each side claiming that it is waiting for an input from the other side. During a `process` command `name` will typically be "Read" or "Wait", the rest of the line contains information about the current state of the RPC protocol. A similar line of output can be found in the Enterprise log.

```
[Waiting for reason_for_wait, wait_time = mmmmm ms]
```

These lines indicate that the main thread of the program is waiting for an event or a resource to become available. Values for `reason_for_wait` include: `TCP/IP input`; `SQL execution to complete`; `bulk loader thread to complete`; `backlog to dissipate`; `index thread to complete`; `mutex name`; and `semaphore name`. These messages are described individually within this section.

```
Log File: "name"
```

This line displays the name of the client log file, which resides in the `logs` folder.

```
Trace: {on, mask = 0xhhhh, trace_file = "name" | off}
```

This line indicates whether tracing is enabled and provides the trace mask value and the name of the trace file, if any.

Verbose: {on | off}, Scheduling: {on | off}

This line shows the status of the verbose option which enables additional log output that gets written to the trace file. The scheduling flag is only meaningful for dbutility. When using the service, the scheduling function is handled by the service.

```
[Client operations inhibited from hh:mm to hh:mm]
```

This line indicates that a blackout period that inhibits client operation between the specified times is present in the Client configuration file.

```
[Stop processing updates on mm/dd/yyyy @hh:mm:ss]
[Stop processing updates at first QPT of AFN dddd]
[Update Worker Thread[n] {waiting for reason_for_wait, wait_time = dddd
| running | terminated}
... ]
```

These lines indicate whether the Client stops processing updates at the first QPT of the specified audit file or at the specified date and time.

Commit ABSN increment will be set to *nnn* at next quiet point

This message is displayed in response to a dbutility COMMIT ABSN *nnnn* command.

```
Commit Parameters: ABSN_inc = aaa, UPDATE_inc = bbb, TIME = ccc, DMS_Txn = ddd
DBserver TXN Group Statistics (cumulative):
Commits = ddd, Avg UPDATE_inc = nnn.nn, Avg Trans time = sss sec, Avg ABSN_inc =
ddd
Rollbacks = ddd, Avg UPDATE_inc = nnn.nn, Avg Trans time = sss sec
```

This message is displayed in response to a dbutility COMMIT STATS command. For details, see [“Process and Clone Commands Messages” on page 66](#).

Commit TIME increment will be set to *nnn* at next quiet point

This message is displayed in response to a dbutility COMMIT TIME *ssss* command.

Commit TXN increment will be set to *nnn* at next quiet point

This message is displayed in response to a dbutility COMMIT TRAN *nnnn* command.

Commit UPDATE increment will be set to *nnn* at next quiet point

This message is displayed in response to a dbutility COMMIT UPDATE *nnnn* command.

Connection to server not yet established

This message indicates that the console operator issued an SSTATS command before the Client established a connection with DBServer.

Console Input: 'text'

The Client logs all console commands issued by the operator, ensuring that there is a record of all such commands. This message, which only appears in the log file, contains the console command text as entered by the operator.

```
Console RPC: {Quit | Quit At hhhmss | Quit After AFN dddd | Quit Now |
Get_Server_Stats | Switch_Log_File | Switch_Trace_File}
```

These messages, which are only applicable to DBClient, are only written to the log file. They provide a log of the console command RPCs that were received by the Client.

DataSource *name* idle

This message indicates that the DBCIntCfgServer program terminated after 1 minute of inactivity. When a new console connection starts up it causes the DBCIntCfgServer program to be started for each of the data sources included in the service's configuration file. This program provides access to the control tables for the console. This message indicates that the DBCIntCfgServer program terminated after a minute of no console activity.

{DBClient | dbutility} will stop after AFN *ddd*

This message, in response to a QUIT (STOP) AFTER *afn* command, indicates that the Client will stop after the given audit file is processed.

{DBClient | dbutility} will stop at *hh:mm:00*

This message, in response to a QUIT (STOP) AT *hh:mm* command, indicates that the Client will stop at the specified time.

{DBClient | dbutility} will stop at the next quiet point

This message, in response to a QUIT (STOP) command, indicates that the Client will stop at the next quiet point.

Log file switched to "*filename*" (Operator Keyin)

This message is displayed in response to a successful LOGSWITCH command, which closes the current log file and starts a new one.

Operator commands: *cmd\_list*

This message displays a list of the available dbutility console commands when a HELP command is issued.

Performance statistics:

This message is printed in response to a dbutility PSTATS command or a Query Performance Statistics command from the Client Console. It is followed by performance statistics that look exactly like the Incremental Statistics outlined in the [Process and Clone Commands \(page 66\)](#) section.

Performance statistics not available

This message indicates that the operator issued a PSTATS command. However, performance statistics are not currently available because the Client has not yet started receiving extracts or updates from the Databridge Engine or Databridge Enterprise Server.

Performance statistics only available during process and clone commands

This message indicates that the operator issued a PSTATS command while executing a command other than process or clone.

Scheduling {disabled | re-enabled}

This message, in response to a SCHED command, indicates that the operator either disabled scheduling, or enabled scheduling that was previously disabled by a SCHED OFF command. This command takes effect only when it is time to schedule the next process command and does not affect the currently executing process command. If scheduling is disabled, the process command simply exits at this point.

Server statistics only available during a process or clone command

This message indicates that the operator issued a SSTAT command during a command other than the process or clone.

Server statistics will be displayed after next quiet point

This message indicates that the server statistics are not available by Client request until the next quiet point. This is the standard response to the SSTATS console command.

```
Server Statistics:
Usercode:         usercode
Priority:         nn
Processor time:   nn.nnnn seconds
I/O time:        nn.nnnn seconds
ReadyQ time:     nn.nnnn seconds
Support version: vv.vvv.vvvv [timestamp]
Support:         (usercode) filespec ...
Filter:          name
```

These messages are the result of the SSTATS command, which requests the server statistics at the next quiet point when the Client is in control of the communications channel to the server. Some of these output lines may be omitted if they are not applicable. *Usercode* shows the USERCODE under which DBServer is run. *Priority* shows the priority of the DBServer worker task. *Processor time* displays the processor time used by the DBServer worker. *I/O time* displays the I/O time used by the DBServer worker. *ReadyQ time* displays the ready queue time used by the DBServer worker; this is time spent when the task is ready to run but cannot get a processor.

**NOTE:** If the `-v` option is enabled, this message will include additional lines of output. These lines are typically suppressed by the Client because the information is redundant.

Statistics function not supported by server

This message indicates that the statistics function is not supported by DBServer because it predates the implementation of RPC currently in use.

Trace file switched to "*filename*"

This message is displayed in response to a successful TSWITCH command, which closes the current trace file and starts a new one with the specified name.

Trace\_options set to 0xddd

This message, in response to a TRACE command, indicates that tracing is now set to the specified value.

```
Update Worker thread[nn] {waiting for wait_condition, wait_time = ssss ms |
running | terminated}
```

These messages, which are part of the STATUS command output when multi-threaded updates are enabled, indicate the state of individual update worker threads. See the preceding "Client State.." entry. If the thread is waiting for an event, the event is shown in *wait\_condition* and *ssss* is the amount of time (in milliseconds) the thread has been waiting.

Possible values for *wait\_condition* include:

- ◆ work which indicates that the thread is idle
- ◆ SQL execution to complete which indicates that the thread is waiting for the database API code to return control to it after the update is completed.

waiting for mutex *name* which indicates that the thread is waiting on a mutex whose name is provided. If this condition does not clear in a reasonable amount of time, the program might be deadlocked.

- ◆ waiting for semaphore *name* which indicates that the thread is waiting for the specified semaphore to be posted. If this condition does not clear in a reasonable amount of time, the program might be deadlocked.
- ◆ running indicates that the thread is actively executing the SQL for an update.
- ◆ terminated indicates that the thread has exited. This only happens when the Client is shutting down.

unknown status *dd* which indicates that the status of the thread is invalid because of an internal error in the program.

Verbose flag set to {true | false}

This message, which is a response to a VERBOSE command, indicates whether the verbose option is set.

Waiting for backlog to dissipate, wait\_time = *mmmm* ms

(Windows only) This message is one of several lines of output produced by the STATUS command. It indicates that the Client has been waiting for the backlog caused by excessive posting for the *bulk\_loader* thread or index thread work queues to dissipate. If this number is very large, it may indicate an internal error in the Client.

Waiting for bulk loader thread to complete, thread\_state = 'state', wait\_time = *mmmm* ms

(Windows only) This message is one of several lines of output produced by the STATUS command. It indicates that the Client has been waiting for the specified amount of time for the *bulk\_loader* thread to complete. If this number is very large, the *bulk\_loader* thread may be blocked waiting for a database resource to become available. The *thread\_state* is a text message that describes what the *bulk\_loader* thread is doing.

Waiting for index thread to complete, thread\_state = 'state', wait\_time = *mmmm* ms

(Windows only) This message is one of several lines of output produced by the STATUS command. It indicates that the Client has been waiting for the specified amount of time for the index thread to complete. If this number is very large, the *bulk\_loader* thread may be blocked waiting for a database resource to become available.

Waiting for mutex *name*, wait\_time = *mmmm* ms

A thread that needs access to a critical section of code must first acquire the mutex. If the mutex is in use, the thread must wait until it becomes available. If a thread is stuck waiting for a mutex, this is usually an indication of a deadlock. In such cases, all you can do is abort the run.

Waiting for semaphore *name*, wait\_time = *mmmm* ms

If a thread needs to execute code after another thread has finished a related task, it will typically do that with a semaphore, unless the associated thread has already posted the semaphore. When the associated thread is done it posts the semaphore. If a thread is stuck waiting for a semaphore, it is usually an indication of a deadlock, which requires that you abort the run.

Waiting for SQL execution to complete [table = 'name'], wait\_time = *mmmm* ms

One of several lines of output produced by the `STATUS` command, this message indicates that the Client has been waiting for the specified amount of time for an executed SQL statement to complete. If this number is very large, execution is blocked in the database server.

Waiting for TCP/IP input from DBServer, wait\_time = *mmmm* ms

One of several lines of output produced by the `STATUS` command, this message indicates that the Client has been waiting for the specified amount of time for input from DBServer. If this number is very large, DBServer is blocked waiting for some event (for example, for the operator to make an audit file available).

## Client Errors

Client errors appear onscreen and in the log file, unless noted.

These messages are prefixed by the timestamp *hh:mm:ss* (where *hh* is hours, *mm* is minutes, and *ss* is seconds). Frequently, errors are preceded by a relational database message. When that occurs, refer to your relational database documentation.

---

**NOTE:** Often, a primary problem will cause several secondary problems, resulting in additional errors. Try to find the earliest error and solve that problem first before proceeding. In many cases, solving this one problem resolves the other problems without any additional work.

---

ERROR: AA value 0xname[/rectype]

This message, which can occur during the data extraction phase of a `process` or `clone` command, indicates that the AA value in the after image of a DMSII link does not match the AA value of the previous CREATE image for the specified data set record. This situation, which should never occur under normal circumstances, indicates a Databridge Engine error.

ERROR: All Engine COMMIT parameters are zero; at least one of them needs to have a positive value

This message, which can occur at the start of a `process` or `clone` command, indicates the resulting commit parameters are in error, as they do not specify any situation under which the Engine should try to do a commit. Since this could result in the system running out of log space or the update process running extremely slowly, the Client will exit with an exit code of 2060 and request that the situation be rectified by updating the configuration file.

ERROR: Ambiguous command *cmd*, matches both *cmd1* and *cmd2* commands

In response to command line console input, this message indicates that the text of the command *cmd*, which the operator typed, matches more than one command. Make sure that you type enough characters to make the command unique.

ERROR: Archive file contains a malformed record: *record\_text*



This message can occur during a `reload` command if the input scanner does not find the record to be in the expected format. If you edited the unload file created by the `unload` command you are likely to get this message or any of the other `reload` command error message. The second line of the message is the record that the program is having a problem with.

ERROR: Archive file "*name*" contains no data or cannot be read

This message can occur during a `reload` command, if an unexpected end of file situation occurs. A possible reason for this would be if the `unload` command that created the file did not complete the command successfully.

ERROR: Archive file does not start with a version record of the form "*V,version*"

This message can occur during a `reload` command, if the file specified on the command line is not a control table unload file. The first record of this file is always "*V,version*" where *version* is the version of the Client control tables.

ERROR: Attempt to ALTER SESSION to set NLS\_LANGUAGE failed

This message, which only applies to the Oracle Client, can occur when the client starts up. It indicates that Client is attempting to execute an `ALTER SESSION SQL` command to set the `NLS_LANGUAGE` to `AMERICAN`, `AMERICA`. This is necessary, as all of the SQL data that the Client creates is formatted to use the period as the decimal character. Since we cannot do this for the `SQL*Loader` runs, we must format the `SQL*Loader` data using the proper decimal character. The Client handles all of this automatically. This message might indicate that the `userid` you are using does not have enough privileges.

ERROR: Attempt to ALTER SESSION to set NLS\_LENGTH\_SEMANTICS failed

This message, which only applies to the Oracle Client, can occur when the client starts up and detects a `AL32UTF8` database. When this occurs, the Client executes an `ALTER SESSION SQL` command to set the `NLS_LENGTH_SEMANTICS` to `CHAR`, which prevents data truncation errors from occurring when 8-bit characters are translated to multi-byte characters. The Client handles all of this automatically. This message might indicate that the `userid` you are using does not have enough privileges.

ERROR: Attempt to clear DAOPT\_Nulls\_Allowed option for key items failed

The Client does not normally allow keys to have the `NULL` attribute. This would invalidate the SQL that is used to perform update and delete operations, where the `WHERE` clauses tests the key items for equality. If the items are `NULL`, you cannot test them for equality. The client attempts to reset the bit that indicates the item allows nulls. This message is displayed in the unlikely situation where the update statement fails.

ERROR: Attempt to connect to a NULL data\_source; add datasource specification to configuration file

This message indicates that there is no `datasource` parameter specified in the [`signon`] section of a client that uses ODBC or CLI. You must specify a `datasource` parameter so that the client can connect to the database using ODBC or CLI.

ERROR: Attempt to delete record from table '*name*' after a duplicate deleted record was unsuccessful (AFN=*afn*, ABSN=*absn*, SEG=*seg*, INX=*inx*) - Keys: *columnname* = *value*, ...

This message can occur during a `process` or `clone` command and it only applies to Miser databases. It indicates that after the Client got a duplicate record while trying to mark a record as deleted, and it then tried to delete the record, the delete resulted in a SQL error. This message may appear after other related error messages that should be addressed first.

ERROR: Attempt to delete record from table 'name' during a modify was unsuccessful (AFN=afn, ABSN=absn, SEG=seg, INX=inx) - Keys: columnname = value, ...

This message can occur when processing a MODIFY record during change tracking. It indicates that, while handling this request as a delete/insert (typically done when the values of keys change), the Client encountered an error during the delete operation. See the database API error that precedes this message to determine the reason why the delete statement failed.

ERROR: Attempt to delete record from table 'name' during a two step modify was unsuccessful (AFN=afn, ABSN=absn, SEG=seg, INX=inx) - Keys: columnname = value, ...

This message, which can occur during a process or clone command, indicates that the Client was unable to delete the record from the specified table during a two-step modify, which is invoked when the value of the depends item for an item with an OCCURS DEPENDING ON clause changes. If the value of this item decreases, the Client updates the rows that remain in the OCCURS table and deletes the rows that are no longer present. This error appears if the delete operation fails and may appear after other related error messages that need to be fixed first.

ERROR: Attempt to delete record from table 'name' was unsuccessful (AFN=afn, ABSN=absn, SEG=seg, INX=inx) - Keys: columnname = value, ...

This message, which can occur during a process or clone command, indicates that the Client was unable to delete the record from the specified table. If deleted records are being preserved the delete operation is actually an update. This message may appear after other related error messages that should be addressed first.

This message can also occur during an insert that results in a duplicate record and the attempt to recover by doing an update results in no rows being updated. The client attempts to recover from this situation by doing a delete/insert. You will get this message if the delete fails.

ERROR: Attempt to delete records from table 'name' during a modify was unsuccessful (AFN=afn, ABSN=absn, SEG=seg, INX=inx) - Keys: columnname = value, ...

This message, which can occur during a process or clone command, indicates that the Client was unable to update a record because it did not exist in the database. When a failed update involves an OCCURS table, the insert statement fails. The Client attempts to recover from the situation by deleting all of the records and reinserting them in the table. This message, which rarely occurs, indicates that the delete failed. This message will be preceded by database error messages that might be helpful in determining why this happened.

ERROR: Attempt to delete records from table 'name' was unsuccessful (AFN=afn, ABSN=absn, SEG=seg, INX=inx) - Keys: columnname = value, ...

This message, which can occur during a process or clone command, indicates that the Client was unable to insert a record into a table and is preceded by several other error messages, warnings, and database error messages that help you determine why this happened. The chain of events leading to this message is:

1. The Client is unable to insert a record into a table because doing so would result in a duplicate record.
2. It attempts to do an update instead but finds no matching rows.
3. When the item has an OCCURS clause, the Client attempts to recover from this situation by doing a delete/insert -- it tries to delete all occurrences of the item and reinsert them.
4. If the delete fails, you get this message.

ERROR: Attempt to drop history table 'name' while "inhibit\_drop\_history" option is enabled

This message, which can occur during a `process` or `clone` command, indicates that the Client attempted to delete a history table while the `inhibit_drop_history` parameter was set to True. This parameter is designed to safeguard against accidentally dropping history tables.

ERROR: Attempt to drop index '*name*' on table '*name*' failed

This message can occur during a `process` or `clone` command when a table that has data that needs to be preserved is reclone. When the records need to be preserved in a nonstandard manner, the program drops the index of the table and runs a cleanup script to delete unwanted records instead of dropping the table and recreating it. The most common source of this error is that the table in question does not have an index, possibly because the index creation during the original clone failed. The program traps this error and continues execution after printing a WARNING. Look at the database error messages that precede this error for clues as to why the drop index operation failed.

ERROR: Attempt to establish IPC connection with DBClntControl failed

This message, which applies to the DBClient and DBClntCfgServer programs, indicates that the IPC connection to the service/daemon could not be established. This connection is used to route output messages to the console via the service/daemon and to allow the console/configurator to issue RPCs to the clients and get the results passed back.

This is an internal error that is an indication that something is seriously wrong with either the DBClntControl program or the system. You should first try to stop and restart the service and if that does not do any good you should consider rebooting the system to clear up the problem

ERROR: Attempt to insert record into table '*name*' after delete was unsuccessful (AFN=*afn*, ABSN=*absn*, SEG=*seg*, INX=*inx*) - Keys: *columnname* = *value*, ...

This message, which can occur during a `process` or `clone` command, indicates that the Client unsuccessfully tried to recover from a failed insert by doing a modify. The client then tries to delete the record or all the occurrences of the given key and tries to reinsert them. This message is an indication that this last insert failed. At this point the client gives up and stops. See the preceding database API error messages for details on why the insert failed.

ERROR: Attempt to insert record into table '*name*' after failed update was unsuccessful (AFN=*afn*, ABSN=*absn*, SEG=*seg*, INX=*inx*) - Keys: *columnname* = *value*, ...

This message, which can occur during a `process` or a `clone` command, indicates that the program was unsuccessful in performing an update because the target record does not exist. The program then tried to do an insert, which also failed because a duplicate was found. A possible cause of this error is that one of the keys is NULL. If this error occurs, contact Micro Focus Technical Support.

ERROR: Attempt to insert record into table '*name*' during a modify was unsuccessful (AFN=*afn*, ABSN=*absn*, SEG=*seg*, INX=*inx*) - Keys: *columnname* = *value*, ...

This message can occur when processing a MODIFY record and the update statement fails to find the record (that is, the row count is 0). The client attempted to recover from this by doing an insert, which failed. Refer to the database API error messages for details on why the insert failed.

ERROR: Attempt to insert record in table '*name*' during a two step modify was unsuccessful (AFN=*afn*, ABSN=*absn*, SEG=*seg*, INX=*inx*) - Keys: *columnname* = *value*, ...

This message can occur during a `process` or `clone` command and indicates that the Client could not insert the record into the specified table during the two-step modify. A two-step modify is used when the value of the depends item changes for an item with an OCCURS DEPENDING on clause. If the value of this item increases, the Client updates the rows that were present in the OCCURS table and inserts the remaining rows. This error appears if the insert operation fails. This message may appear after other related error messages that should be addressed first.

```
ERROR: Attempt to insert record in table 'name' was unsuccessful
```

This message, which can occur when a data set is cloned without using the bulk loader. It indicates that an error occurred when the client tried to do a COMMIT after `max_clone_create` count records were inserted into the table. See the preceding database API error messages for details on why the commit failed.

```
ERROR: Attempt to insert record into table 'name' was unsuccessful - Keys:
columnname = value, ...
```

This message can occur when a data set is cloned without using the bulk loader. It indicates that an error occurred when the Client tried to insert the record into the table. Refer to the database API error messages for details on why the insert failed.

```
ERROR: Attempt to insert record into table 'name' was unsuccessful (AFN=afn,
ABS=absn, SEG=seg, INX=inx) - Keys: columnname = value, ...
```

This message, which can occur during a `process` or `clone` command, indicates that the Client was unable to insert the given record into the table. The Client always attempts to recover from a failed insert that is caused by the record already being in the database. If we get a different error, then this message is displayed and the client stops. Refer to the database API error messages for details on why the insert failed.

```
ERROR: Attempt to load record into table 'name' was unsuccessful - Keys:
column_name = value, ...
```

This message, which is limited to the SQL Server Client, indicates that an error occurred while setting up the host variables during data extraction or that the BCP API's `bcp_sendrow` procedure returned an error. In either case, look at the message that precedes this one in the log file to help identify the actual reason for this error.

```
ERROR: Attempt to mark record as deleted in table 'name' failed (AFN=afn,
ABS=absn, SEG=seg, INX=inx) - Keys: columnname = value, ...
```

This message indicates that the `process` or `clone` command (used in conjunction with the delete record preservation feature) was unable to update a record and set the appropriate column to indicate that it has been deleted. Refer to the SQL error messages for clues about why this error occurred.

```
ERROR: Attempt to reclone DataSet name[/rectype] without recloning DataSet name1
[/rectype1]
```

The `clone` command is attempting to reclone the specified data set and the following conditions exist:

- ◆ The configuration parameter `automate_virtuals` is set to `True`.
- ◆ The specified data set is the primary source for a virtual data set that gets its input from more than one DMSII data set.
- ◆ The data set that is the secondary source of data has been cloned previously and is not specified on the command line.
- ◆ Recloning is only allowed for the data set that is the secondary source of the data. For example, assume `SV-HISTORY` and `SAVINGS` both provide data for the virtual data set `SV-HISTORY-REMAP`, and `SV-HISTORY` is the primary source of data for `SV-HISTORY-REMAP` that must be cloned first. An attempt to reclone `SV-HISTORY` without recloning `SAVINGS` results in this error. If you want to reclone `SV-HISTORY`, you must also reclone `SAVINGS`. If you specify them both on the command line, the program performs operations in the right order. However, you can reclone `SAVINGS` without recloning `SV-HISTORY`.

ERROR: Attempt to rename table '*name*' failed

This message can occur during the execution of a `reorg` command. It indicates that an error occurred while attempting to rename a table. When the parameter `use_internal_clone` is set to `True`, the `redefine` command sets up the `reorg` scripts to use a `SELECT INTO` statement (CTAS in the case of Oracle) to create a new copy of the table that has the new layout. Added columns are assigned their initial values. Before running the script that does the copy, the `reorg` command executes a SQL statement that renames the old table. You will see this message when the rename fails. If the `reorg` script that does the copy fails, the new table is dropped and the old table is renamed back to its original name. You will also see this message if the rename fails in this case.

ERROR: Attempt to run user stored procedure '{*m\_tablename* | *r\_tablename*}' failed

This message, which only applies to MISER databases, can occur during a `process` or `clone` command and indicates that an error occurred while running the stored procedure. This message may appear after other related error messages that need to be fixed first.

ERROR: Attempt to treat duplicate insert record as an update for table '*name*' failed (AFN=*afn*, ABSN=*absn*, SEG=*seg*, INX=*inx*) - Keys: *columnname* = *value*, ...

The Client traps all duplicate record errors that occur during the execution of insert SQL statements. It then changes the insert statements into update statements and re-executes them. This error is displayed when this happens while processing an update as a delete/insert. If the resulting update statement fails, this error message is displayed because it cannot find the target record to change. If this error occurs, contact Technical Support.

ERROR: Attempt to update table '*name*' was unsuccessful (AFN=*afn*, ABSN=*absn*, SEG=*seg*, INX=*inx*) - Keys: *columnname* = *value*, ...

This message, which can occur during a `process` or `clone` command, indicates that the Client could not update the specified table. This message may appear after other related error messages that need to be fixed first. This error does not apply to situations where the Client encounters an update with no matching rows; the Client automatically recovers from that situation.

ERROR: Bad bit (0x*hhhhhhh*) in {DataItem | DataSet | DataSource | DataTable | DMSItem} column mask

This message occurs when DBCIntCfgServer attempts to update an Client Control Table entry and gets an invalid bit mask. This is an internal error in the Client Console or the Configurator. Contact Micro Focus Technical Support.

ERROR: Bad data mask string index, column 'name' in table 'name' cannot be masked

This message, which is limited to the SQL Server Client, indicates that the data mask specified for the given column is incorrect. Data masks, which are stored in the `masking_info` column of the DATAITEMS Client control table, are constructed as 32-bit integers, where the low 8 bits are the masking function code (0 indicates no masking, 1 indicates "default" masking, 2 indicates "email" masking, 3 indicates "random" masking and 4 indicates "partial" masking). The rest of the value is the index into the `masking_parameter` array in the client configuration file, where these values must be specified.

The Client uses this information to add a specification of the form "masked with (function=name([parameters]))". Only the random and partial functions have parameters. The random function can be applied to numeric columns, while partial function can be applied to columns of char or varchar type. This error indicates that the specified string is in error.

ERROR: Bad {index | table} suffix value *nn* specified for table 'name' in DATATABLES

This message occurs during a `generate` or `reorg` command if the `index_suffix` or `create_suffix` column of the corresponding DATATABLES entry has a value that is out range or there is no [*n*] specification in the configuration parameters `create_table_suffix` or `create_index_suffix`.

ERROR: Bad input line 'text' in globalprofile.ini

This message, which is limited to UNIX clients, indicates that file `globalprofile.ini` contains a bad input line. Do not add comments or any other lines to the file, as this will most probably cause this error.

ERROR: Bad section header 'name' in globalprofile.ini

This message, which is limited to UNIX clients, indicates that first line of file `globalprofile.ini` has been modified. The first line of this file must contain the section header [`dbridge`].

ERROR: `bcp_bind` failed for column 'name' in table 'tablename'

This error message that only applies to the SQL Server client indicates that the client was unable to bind the specified column to host variables while setting up the BCP API to load the table in question. Contact Micro Focus Customer Care if you get this error.

ERROR: `bcp_init` failed for "database..table"

This error message that only applies to the SQL Server client indicates that the call on the BCP API's `bcp_init` procedure failed. Check the database name and the table name and then Contact Micro Focus Customer Care if you get this error.

ERROR: `bcp_sendrow` failed for table 'tablename'

This error message that only applies to the SQL Server client indicates that the call on the BCP API's `bcp_sendrow` procedure failed. Check the ODBC error that is associated with this error as there might be a simple explanation such as the database is out of storage space.

ERROR: `Begin_Transaction` during data extraction for table 'name' not using the bulk loader failed

This error message, which is limited to data extractions that do not use the bulk-loader (or the BCP API). Such extractions are handled by separate data connections that use transactions to limit the number of updates that are done before doing a COMMIT. Begin transaction should not fail, as it does not do much other than disabling auto-commits. The size of the transaction is controlled by the configuration parameter `max_clone_count`.

ERROR: Binary filter generation failed with exit code *nnnn*

This message indicates that the **makefilter** run that the client launched failed. The client launches **makefilter** at the end of a define or redefine command and when exiting the Client Configurator. You need to fix the error in the filter source and rerun **makefilter** to fix this situation. There is no need to rerun the command, as it completed successfully.

ERROR: Binding of variables failed for SQL statement *sql\_stmt*

This message is only applicable when using host variables, which are enabled by setting the configuration parameter `aux_stmts` to a non-zero value. This indicates that an internal error occurred while binding program variables to various columns of an SQL statement. In this mode, all SQL statements use bound variables. Before the SQL statements are actually executed, the values are copied into these variables by the program. This technique allows the SQL statement to be re-executed using different data since the SQL remains constant. Only the content of the host variables changes.

ERROR: Bulk copy record count (*mmm*) and actual record count (*nnn*) differ for table '*name*'

This message, which is only displayed if the configuration parameter `verify_bulk_load` is set to 2, indicates that there is a mismatch between the number of rows in the table and the number of rows that the client loaded. Look at the bulk loader log file for the table in the working directory and also in the discards directory where you will undoubtedly find a discard file for the given table. If the parameter `verify_bulk_load` is set to 2, this situation causes the client to abend. However, if it set to 1, you only get a similar warning, and the client ignores the error.

ERROR: Bulk load failed (rc = *dd*) for table '*name*' using *sql\*loader*  
- see file "*sqlld.tablename.log*" for more information

This message indicates that there was a problem loading the extract records to the corresponding database table using the bulk loader. The first place you should look is the bulk loader log file in the working directory. This will usually indicate what the problem is. The most common problems are:

- ◆ The maximum error threshold was exceeded causing the bulk load to abort the operation.
- ◆ The database's bin directory is not in the PATH causing the attempt to run the bulk loader to fail.
- ◆ The database or table space is out of space.

In the case of the Microsoft SQL Server Client the sequence (rc = *d*) specifies the exit code of the launched command file that does the bcp. An exit code of 1 indicates that bcp returned a non-zero exit code and a return code of 2 indicates that `bcp_auditor` utility determined that the bcp failed, even though bcp returned an exit code of 0.

ERROR: Cannot access Databridge control tables  
-Make sure that a 'configure' command was executed previously

This message can occur for any Client command except `dbutility configure`. It can indicate either of the following:

You did not run the `dbutility configure` command and therefore the client control tables were not created.

—or—

(more common) You are trying to execute a `dbutility` command with a relational database user ID that is different from the one that originally created the Client control tables. In this case, use a relational database query tool to sign on to relational database with the *same user ID that `dbutility` is using*. Then, enter the following:

For Microsoft SQL Server:

```
select name, uid from sysobjects where type = 'U'
```

For Oracle:

```
select table_name from user_tables
```

The resulting display includes the name of all of the user tables owned by the user ID. If all of the Client control tables do not appear, the problem is in the user ID you were using. For more information, see the *Databridge Installation Guide*.

```
ERROR: Cannot access user# for user userid
```

(SQL Server Client only) When you sign on to the relational database, the Client reads the Microsoft SQL Server table `sysusers` to find the user index for your user ID. The user index is important because the relational database uses the user index (numeric) rather than the user ID (character string) to mark table ownership.

When this error occurs, it is typically preceded by messages that help to illustrate what went wrong. If this error persists, contact your relational database administrator.

```
ERROR: Cannot find entry point "EBCDIC_to_ASCII" in extended translation {DLL | shared library} "filename"
```

The attempt to locate the entry point `EBCDIC_to_ASCII` in the data translation DLL failed. Make sure that there is not another DLL with the same name that Windows is finding instead. The best way to avoid this problem is to put the DLL in the Databridge Client program directory and setup the `PATH` environment variable to include this directory. Because the Client is executed from this directory, and the DLL is also present in that directory, you should not run into this problem.

```
ERROR: Cannot open lock file for data source name, errno=number (errortext)
```

This message indicates that the Client cannot open the lock file used to implement a file lock for the data source. A common cause of this error is that the locks directory does not exist. If you did not use `migrate` to do the update, you must create the service's (global) working directory and the locks sub-directory in order to be able to run the client. For more information, see the *Databridge Installation Guide*.

```
ERROR: Cannot run a cmd_name command while the data source is locked
```

This message indicates that a Databridge run has the data source locked. When a run hangs, you must cancel the run to release the lock file. The `process_id` of the run that locked the file and the command it is executing is provided in the lock file `source_database.lock` (where `source` is the name of the data source and `database` is the name of the relational database). This file is located in the `locks` subdirectory of the service's (global) working directory.



ERROR: chdir failed for directory "*path*", errno=*number* (*errortext*)

This message, which can occur when processing the `user_script_dir` line of a text configuration file or when updating this parameter in a binary configuration file. It can also occur when running a user script. When file security is enabled, user scripts must reside inside the service's working directory, as they will then be protected by file security. When the client needs to verify that a user script or the specified user script directory complies with this rule, it does a couple of `_chdir` and `_getcwd` commands while trying to verify this. This error is an indication that a directory is missing or that the parameter `user_script_dir` is pointing to a non-existent directory.

ERROR: CheckTokenMembership() failed for group '*name*', errno=*number* (*errortext*)

(Windows only) This message, which only applies when file security is enabled, indicates that the system call to verify the user's membership in a group failed. The client uses this call to determine if the user is allowed to run the client, by determining if the user has write permissions for the files in the working directory.

ERROR: Cleanup of table '*name*' failed

The Client was unable to delete selective entries from the specified table during the `drop` or `dropall` command. The Client typically drops tables. However, if tables contain non-DMSII data or are populated from more than one data source, the Client uses the cleanup scripts to delete the records in question. Look at the relational database messages that precede this error for more information.

ERROR: Client aborting *rpc\_description* call

This message is preceded by other messages such as relational database API errors. The *rpc\_description* indicates the specific operation (such as `connect`, `initialize`, `switchaudit`) that was in progress when the Client error occurred. This message indicates that a fatal error occurred while processing the given DBServer RPC.

ERROR: Client Control Table version mismatch (found *number*, expected *number*)  
- Run the "dbfixup" program to correct this situation

(dbutility only) This message indicates that the client control tables have the wrong version. You must run the `dbfixup` program to correct this error. This message typically occurs if you upgrade from an earlier version of the Client and do not run the `dbfixup` program before you run the Client. If you use the `Migrate` program, it automatically runs `dbfixup`.

ERROR: Client Control Table version mismatch (found *number*, expected *number*)  
- The {*service* | *daemon*} will automatically run "dbfixup" to correct this situation

This message indicates that a client launched by the service has detected that the client control tables have the wrong version. The service automatically launches the `dbfixup` program when this happens. The console is usable as soon as the data source turns green (it is marked as off-line during the execution of `dbfixup`). The properties of data source will be missing until you do something that causes them to be refreshed.

ERROR: Client/host interface level mismatch: client = *ver1*, host = *ver2*

This message can occur when the Client connects to DBServer, if the version of DBServer is too old to be compatible with the Client. Specifically, this message indicates that DBServer returned an illegal value for the negotiated protocol level. We recommend that you always use matching host, Enterprise Server, Client, and Console software.

ERROR: Close failed for binary configuration file "*name*", errno=*number* (*errortext*)

This is an indication that there was a problem writing the binary configuration file to disk. The *errortext* explains the cause of the failure. The most common cause is that the `config` directory has not been created or you do not have write access the `config` directory or the file `dbridge.cfg`.

ERROR: Close failed for Null Record file, *errno=number* (*errortext*)

An error occurred while closing the NULL record file during a `define` or `redefine` command. The most likely problem is a lack of disk space. The system error included in the message should explain why the error occurred.

ERROR: Close failed for text configuration file "*name*", *errno=number* (*errortext*)

This is an indication that there was a problem writing the text configuration file to disk. The *errortext* usually indicates the cause for the failure. The most common cause is that the `config` directory has not been created.

ERROR: Column '*name*' in table '*name*' is not an AA Value -- this is not supported

This message only occurs what you clone embedded subsets and the client tries to execute a bulk delete operation which deletes all the child record that belong to given parent in the virtual data set tables that implements the embedded subset. This message is indicative of a configuration error as the parent record must be using AA values as the index; without that the parent/child relationship cannot be implemented.

ERROR: Command aborted due to error in cross checking DMSII Links

The `define` or `redefine` command is being aborted because the Client encountered errors when checking the integrity of DMSII links. This message is preceded by one or more error messages indicating which tables have links to nonexistent or inactive tables. The most likely cause of this error is that you are attempting to do a partial `redefine` of the data source with a data set that is the target of a link not being included in the `redefine` command.

ERROR: *command\_name* command failed to complete

This message indicates that the Client command could not complete successfully, where *command\_name* is a client command such a `process`, `clone` or `redefine`. Typically, this message is preceded by an explanatory error.

ERROR: COMMIT ABSN command requires a numeric argument

This message is a result of an invalid user input for a `dbutility` console `COMMIT ABSN` command.

ERROR: COMMIT ABSN value out of range

This message is a result of an invalid user input for a `dbutility` console `COMMIT ABSN` command.

ERROR: COMMIT command must be followed by {Absn <n> | Update <n> | TIme <n> | TRans <n> | Stats}

This message is a result of an invalid user input for a `dbutility` console `COMMIT` command.

ERROR: COMMIT TIME command requires a numeric argument

This message is a result of an invalid user input for a `dbutility` console `COMMIT TIME` command.

ERROR: COMMIT TIME value out of range

This message is a result of an invalid user input for a `dbutility` console `COMMIT TIME` command.

ERROR: COMMIT TRANS command requires a numeric argument

This message is a result of an invalid user input for a dbutility console COMMIT TRANS command.

ERROR: COMMIT TRANS value out of range

This message is a result of an invalid user input for a dbutility console COMMIT TRANS command.

ERROR: COMMIT UPDATES command requires a decimal argument

This message is a result of an invalid user input for a dbutility console COMMIT UPDATES command.

ERROR: COMMIT UPDATES value out of range

This message is a result of an invalid user input for a dbutility console COMMIT UPDATES command.

ERROR: Commit\_Transaction during data extraction for table 'name' not using the bulk loader failed

This error message is limited to data extractions that do not use the bulk-loader (or the BCP API). Such extractions are handled by separate data connections that use transactions to limit the number of updates that are done before doing a COMMIT. The size of the transaction is controlled by the configuration parameter `max_clone_count`. The failure of the COMMIT is most likely a database resource issue. The preceding database API error messages should help in determining the source of the problem.

ERROR: Configuration file "name" contains no valid information

This message indicates that the specified configuration file contains no data or valid information (for example, there are no section headers in the file).

ERROR: Configuration file [name] section: *error\_message*  
- Input Line: *input\_text*

This message is associated with a large number of errors that can appear when processing a text configuration file. If you use binary configuration files, these messages are confined to the `import` command.

[*name*] is the relevant section in the configuration file. *input\_text* is the actual line of text in which the error occurs. For information about configuration file syntax, see Appendix C in the *Databridge Client Administrator's Guide*.

ERROR: Configuration file contains invalid line: *input\_text*

This message typically appears when you have omitted the semicolon (;) from a comment line that precedes the first section header in the configuration file.

ERROR: Configured numeric date format (*number*) for item 'name' in table 'name' is not supported, {record will be discarded | date set to NULL}  
-Keys: *columnname* = *value*

This message can occur during a `process` or a `clone` command when processing a data item whose `sql_type` column contains a value of 13 (*numeric\_date*). The configuration parameter `numeric_date_format` is used to define the format for numeric dates. This error indicates that the format specified in the configuration file is not supported.

ERROR: CreateDirectory failed for directory "name", error=*number* (*errortext*)

(Windows only) This message indicates that attempt to create a directory failed, the *errortext* is the explanation of the error and it should help identify the cause of the error.

You should not try to change file security, except by running `setfilesecurity` program. This program is located on the root of the install directory.

ERROR: CreateFile failed for 'Console\_Reader\_Thread'

(Windows only) This message is an internal error, which indicates that the console thread's attempt to read input from the keyboard failed. As a result, the console is inoperative. Except for this, the run proceeds as if the console were not enabled.

ERROR: CreateFile failed for data file "*name*", error=*number* (*errortext*)

(Windows only) This message can occur during a `process` or `clone` command. It indicates that the Client cannot create the specified data file for hold bulk loader data. The indicated system error should explain the cause of the problem.

ERROR: CreateFile failed for file "*name*", error=*number* (*errortext*)

(Windows only) This message can occur when opening a new file for write and file security is enabled. The client creates the new file using the Windows library `CreateFile` procedure, which uses a DACL that is constructed to reflect the access rights based on the security defined at install time. The error will most likely indicate that the user does not have the proper privileges to create the file. Another advantage of using file security is that it allows the service and the command line to share files when the service is run using the built-in SYSTEM account.

ERROR: CreateMutex failed for '*name*', error=*number* (*errortext*)

(Windows only) This message, which can occur during a `process` or `clone` command, indicates that a Windows internal error has occurred while attempting to create a mutex resource. If this error occurs, contact Technical Support.

ERROR: CreateSemaphore failed for '*name*', error=*number* (*errortext*)

(Windows only) This message, which can occur during a `process` or `clone` command, indicates that a Windows internal error occurred while attempting to create a semaphore resource used by the table creation thread. If this error occurs, contact Micro Focus Technical Support.

ERROR: CreateThread failed for '*name*', error=*number* (*errortext*)

(Windows only) This message, which can occur during a `process` or `clone` command and indicates that an internal system error occurred while attempting to create a thread (`bcp_launcher`, `index_creator` or console thread). If this error occurs, contact Micro Focus Technical Support.

ERROR: Creation of control table *name* failed

This message can occur during a `dbutility configure` command or when the control tables are created by the Console or the Client Configurator. It indicates that an error occurred while creating the specified Client control table. See the relational database API message that precedes this message (on the screen or in the log file) for more information.

See also [“OCI Errors” on page 157](#) or [“ODBC Errors” on page 156](#).

ERROR: Creation of history table '*name*' failed

This message, which can occur during a `reorg` command, indicates that the command was unable to create the given history table. The configuration parameter `enable_dynamic_hist` allows the Client to dynamically create history tables without having to reclone the data set in question. See the database API error messages that precede this error message to determine why the creation of the table failed.

ERROR: Creation of keys failed for primary table '*name*'

This message can occur during a `define` or `redefine` command. It indicates that an error occurred while the Client attempted to define the keys for a primary table. This message is preceded by another error message that explains the actual cause.

ERROR: Creation of keys for OCCURS failed for table '*name*'

This message, which can occur during a `define` or `redefine` command, indicates that an error occurred while trying to insert the keys for an OCCURS table into the DATAITEMS Client control table. See the relational database API message that precedes this message (on the screen or in the log file) for more information.

See also [“OCI Errors” on page 157](#) or [“ODBC Errors” on page 156](#).

ERROR: Creation of keys within split failed for table '*name*'

This message can occur during a `define` or `redefine` command. It indicates the following:

- ◆ A DMSII data set has more columns than the relational database limit, and therefore, must be split into two or more relational database tables.
- ◆ An error occurred while creating the keys for a secondary table.

See the relational database API message that precedes this message (on the screen or in the log file) for more information. See also [“OCI Errors” on page 157](#) or [“ODBC Errors” on page 156](#).

ERROR: Creation of table '*name*' and its procedures failed

This message can occur during the cloning of a data set using a `process` or `clone` command if the Client is unable to create a table and its stored procedures. See the relational database message that precedes this error for more information about what went wrong.

ERROR: Critical columns missing from table '*name*', cannot create cleanup script

This message indicates that the `generate` command could not find some of the columns that are created by setting the `external_columns` column of the DATASETS Client control table to a nonzero value. The most common cause of this message is that the `external_columns` column was assigned a value without running a `define` or a `redefine` command, or that the active column for the item in question was set to zero in DATAITEMS. It is recommended that you use user scripts to perform all such actions that the `redefine` command runs and that you never change the value of the active column in DATAITEMS.

ERROR: Data item number *nnn* referenced in filter file does not exist for table '*name*'

This message is an internal error that indicates the binary filter file is not in sync with the client control tables. Try recreating the filter by either running the **makefilter** utility's `import` command or run a `redefine` command with the `-R` option (Redefine All) to remedy this situation. This situation should never occur, as whenever you run a `redefine` command or you run the Client Configurator **makefilter** is automatically launched. However, you need to pay attention to the cases where the program is unable to compile the filter due to errors in the filter source file. Look in the **makefilter** log file to determine why the compile failed.

ERROR: Data source does not support transactions (SQL\_TXN\_CAPABLE = NONE)

This message, which is applicable to ODBC and CLI clients, indicates that the database does support transactions. You cannot run the Databridge Client with a database that does not support transactions. This error indicates that your relational database is not properly set up.

ERROR: Database {*name* | NULL} failed to open [twice]

The requested relational database failed to open. (If you do not specify a database name and you are connecting to the default database, the Client displays "NULL" as the database name.)

In this case, check the following:

- ◆ Is the relational database server running?
- ◆ Did you enter the correct relational database server name?
- ◆ Did you enter the correct relational database name?
- ◆ Did you enter the correct ODBC data source name?
- ◆ Did you use the correct user ID for the relational database?
- ◆ Did you use the correct password?

Check your settings for environment variables, `dbridge.cfg` configuration file, and command line options.

The Client establishes two connections with the relational database server. This error indicates a failure on the second connection when the word `twice` is present.

ERROR: Database password cannot be decoded

This message can occur when the Client reads a text configuration that contains an encoded password that is corrupt. This error happens most often when the file has been edited. To resolve the problem, replace any passwords and then use `dbpwenc` or the `export` command to encode them.

ERROR: Databridge call failed for *rpc\_description* (Transport or RPC error)

This message is preceded by a `SOCKETS ERROR` or other message that occurs during the processing of a remote procedure call. *rpc\_description* indicates the specific operation that was in progress when the Client error occurred (for example, `connect`, `primary_set`, `set_option`, `data_sets`, `initialize`, `switchaudit`). This message can occur in any command that establishes communications with DBServer.

ERROR: Databridge control tables are not empty, use `dropall` command first  
- To bypass this check, use the 'u' option for the `configure` command

This message indicates that you are attempting to run a `dbutility configure` command that overwrites the existing client control tables. This message is intended as a safeguard so that you do not accidentally overwrite the existing client control tables. For information about `-u` and other `dbutility` command options, see Appendix B in the *Databridge Client Administrator's Guide*.

ERROR: DataSet List specification not allowed for a DataSource that is not defined

This message occurs during a `reload` command when you specify a list of data sets to reload for a data source that is not found in the client control tables. The partial reload of only a few specific data sets is only supported if the data source exists and contains entries for the specified data sets.

ERROR: DataSet List specification not allowed when loading all DataSources

This message occurs during a `reload` command when you specify a data source name of `_all` and a list of data sets to reload. The partial reload of only a few specific data sets is only supported if the data source is explicitly named, it exists and it contains entries for the specified data sets.

ERROR: DataSet *name* not found

This message appears during a `clone` or a `refresh` command when one of the following occurs:

- ◆ You did not enter the correct DMSII data set name on the command line.
- ◆ The active column for this data set is set to 0 in the DATASETS Client control table.

ERROR: DataSet *name*[/*rectype*] does not have history tables; DSOPT\_HistoryOnly bit must be 0 in `ds_options`

This message, which can occur during a `define` or `redefine` command, indicates that the `ds_options` column of the DATASETS table entry for the data set is incorrect. The `DSOPT_HistoryOnly` bit can only be set when the `DSOPT_Save_Updates (8)` bit is also set. This is an indication that your user scripts are incorrect.

ERROR: DataSet *name*[/*rectype*] failed reorganization, correct the error or `reclone - mode dd`

This message can occur at the start of a `process` or `clone` command if the Client finds a data set whose mode is 33. This situation can occur if you attempt to run a `process` or `clone` command after running a `reorg` command where the reorganization of a table fails for the data set in question.

To resolve this problem, do one of the following:

Set the `ds_mode` to 0 and `reclone` the data set.

—or—

Fix the reorganization script and rerun the `reorg` command after you set the `ds_mode` to 31 for the data set in question.

ERROR: DataSet *name*[/*rectype*]has an invalid mode *dd*

This message occurs during a `process` command (at data set selection time), if the `ds_mode` column of the Client control table contains an illegal value *dd*. Use a relational database query tool to enter a valid *dd* value.

ERROR: DataSet *name*[/*rectype*]has an invalid value (*dd*) in the `status_bits` field

This message occurs at the start of a `process` or a `clone` command (at data set selection time), if the `status_bits` column of DATASETS Client control table contains an illegal value *dd*. Use a relational database query tool to enter a valid *dd* value.

ERROR: DataSet *name*[/*rectype*] has been reorganized (mode = *dd*); you must first run a `reorg` command

This message can occur at the start of a `process` or `clone` command (at data set selection time), if the Client finds a data set whose mode is 31 or 34. This situation can occur if you attempt to run a `process` or `clone` command after running a `redefine` command that needs to be followed by a `reorg` command.

After you inspect the `reorg` scripts to make sure that the actions they are about to perform are reasonable, run a `reorg` command. `Reorg` scripts ALTER tables, which is nearly impossible to reverse.

ERROR: DataSet *name*[/*rectype*] has been reorganized; you must first run a `redefine` command

This message can occur during a `process` or `clone` command (at data set selection time), if the DBSelect RPC call returns a status indicating that the data set has been reorganized. This situation can occur if you attempt to clone a data set that was defined before the data set was reorganized in DMSII. Simply run a `redefine` command and (if necessary) a `generate` command before attempting the clone again.

ERROR: DataSet *name*'s {*real\_ds\_num* | *virtual\_ds\_num*} column points to an inactive or non-existent structure number

This message only applies when the `automate_virtuals` parameter is set to True. It indicates that the `real_ds_num` or `virtual_ds_num` column for the specified data set points to an inactive or non-existent structure. The proper handling of virtual data sets that get input from more than one real data set requires that all data sets involved have their active columns set to 1. Use a relational database query tool to correct this situation.

ERROR: DataSource *name* already defined, use `redefine` command instead  
- To bypass this check, use the 'u' option for the `define` command

This message occurs if you attempt to run a `define` command for a data source that is already defined. It is a protection against inadvertently running a `define` command when you meant to run a `redefine` command. If you intended to use a `define` command, use the `-u` option on the command line.

ERROR: Day of year value *val* out of range for item '*name*' in table '*name*'. {record will be discarded | date set to NULL} - *Keys:columnname = value,...*

This message can occur during a `process` or `clone` command. It indicates that the specified day value is incorrect in a MISER or LINC database date or in a Julian date. In the case of LINC or Miser dates, this should never happen. If it does, contact Technical Support. In the case of a Julian date, the value in the DMSII database is in error (for example 400).

This is not a fatal error. The date is stored as NULL or if the item is a key, the record is discarded.

ERROR: Definition of data items for table '*name*' failed

This message can occur during a `define` or `redefine` command. It indicates that a failure occurred while defining the data item for the specified table. For more information on the cause, refer to the messages that preceded this one onscreen or in the log file.

ERROR: Delete of *control\_tablename* entries for *datasource* failed

This message can occur during a `define`, `redefine`, `drop`, or `dropall` command. It indicates that the records in the given Client control table could not be deleted. See the relational database API message that precedes this message (on the screen or in the log file) for more information.

See also [“OCI Errors” on page 157](#) or [“ODBC Errors” on page 156](#).



ERROR: Delete\_all for table 'name' failed

This message indicates that the processing of a DELETE\_ALL request failed. This request is used while updating embedded subsets. See the relational database API message that precedes this message (onscreen or in the log file) for more information about why the delete statement failed.

ERROR: Deletion of records from client control tables failed for DataSource name

This message can occur during the define command, it is always preceded by more specific error messages. It is an indication that the define command could not delete the entries that existed in the client control tables. See the database API message that preceded this message for more information.

ERROR: DeSelect for DataSet name[/rectype] ignored, it has no structure index assigned

This message can occur during a process or clone command if the program encounters an internal error when attempting to deselect a data set. This operation is used when the AA values of a data set are invalidated by a DMSII garbage collection reorganization. When a search of the table that contains the selected data sets fails for the data set to be deselected, this message is displayed.

ERROR: DIOPT\_Store\_as\_GUID option only valid for ALPHA(36) items -- option ignored

(SQL Server Client only) This message occurs when you try to set the bit in question for a DMSII item that is not an ALPHA(36). Setting this bit causes the client to treat the data as a GUID and setup the column to have a data type of UNIQUEIDENTIFIER.

ERROR: DMSII database timestamp does not match value in control tables, further processing is not possible

The column db\_timestamp, which was recently added to the DATASOURCES client control table, holds the DMSII database's timestamp. This is normally filled in when the data source is defined and is used to make sure that client is using the same database as when the data source was created. If the column is 0, the test is not performed. The client will update this column in this situation. If you need to bypass this error, simple set the db\_timestamp column, which is binary, to all zeroes.

ERROR: DMSII date contains an illegal numeric value val for item 'name' in table 'name', {record will be discarded | date set to NULL} - Keys: columnname = value, ...

This message indicates that dbutility encountered an illegal numeric value while processing the item which is being interpreted as a DMSII date. This error message applies to both numeric and alpha dates. It can be caused by a number that is longer than 8 digits; a value that contains illegal digits; an incorrect value in the dms\_subtype column; or bad DMSII data. This is not a fatal error. The date is stored as NULL or, if the item is a key, the record is discarded.

ERROR: DMSII date/time contains an illegal time value tval for item 'name' in table 'name', {record will be discarded | date set to NULL}  
- Keys: columnname = value, ...

This message indicates that the Client encountered an illegal numeric value while processing the item which is being interpreted as a DMSII time. This error message applies to both numeric and alpha times. It can be caused by a number that is longer than 6 digits; a value that contains illegal digits; an incorrect value in the dms\_subtype column; or bad DMSII data. This is not a fatal error. The time is stored as NULL or, if the item is a key, the record is discarded.

ERROR: Drop of control table name failed

This message can appear during `dbutility configure` or `dropall` commands. It indicates that the drop of the specified Client control table, data table, or data table's associated stored procedures has failed. See the relational database API message that precedes this message (on the screen or in the log file) for more information.

See also [“OCI Errors” on page 157](#) or [“ODBC Errors” on page 156](#).

ERROR: Drop of table '*name*' failed

This message can appear during a `drop`, or a `dropall` command. It indicates that the drop of the specified data table, or data table's associated stored procedures, has failed. See the relational database API message that precedes this message (on the screen or in the log file) for more information. See also [“OCI Errors” on page 157](#) or [“ODBC Errors” on page 156](#).

ERROR: Drop of table '*x\_name*' failed

This error can occur during a `reorg` command when the parameter `use_internal_clone` is set to True. It indicates that after the `reorg` command created a new copy of the table, the attempt to drop of the old table, which was previously renamed, failed.

ERROR: `email()` masking function is illegal for item '*name*' in table '*name*' data type = *dtype*

This error message, which is limited to the SQL Server client, indicates that an attempt was made to mask the specified column using a masking function of "email". This function is only valid for columns whose data type is char or varchar.

ERROR: Embedded DataSet *name[/rectype]* cannot be selected because its parent structure is not active

This message, which can occur during a `process` or `clone` command, indicates that the Client encountered an embedded data set whose parent structure is not selected. Set the active column of the parent structure to 1 in the DATASETS Client control table.

ERROR: Engine did not send BI/AI pair for update to DataSet *name[/rectype]* which allows key changes - Clear bit 0x1000 in `ds_options` of DATASETS entry if you wish to ignore this error

This message indicates that the Databridge Engine did not send updates as BI/AI pairs as the Client requested. Clear the bit, which is displayed as a hexadecimal mask. This situation can occur if you have bad user scripts that force the data set to use AA Values or RSNs by improper methods. You should set the `DSOPT_Use_AA_Only` (0x800) bit in `ds_options` to do this. If you use the deprecated bit `DSOPT_Include_AA` (16) and clear the `item_key` columns you will get this error.

**CAUTION:** Clearing this bit is inadvisable if the key changes are actually taking place.

ERROR: Execution of script "*filename*" failed

This message can occur during a `reorg` command and indicates that the execution of a `reorg` script failed. See the relational database API message that precedes this message (on the screen or in the log file) for more information. See also [“OCI Errors” on page 157](#) or [“ODBC Errors” on page 156](#).

ERROR: Expected LINK\_AI record for DataSet *name[/rectype]* not received from Server

This message indicates that a Databridge Engine error occurred during the data extraction of a data set that has DMSII links. The data extraction for such data sets consists of CREATE images followed by LINK\_AI records. The Client combines these before writing the resulting record to the bulk loader file (or pipe, on UNIX). If a CREATE record is followed by another CREATE record this error is issued. Contact Micro Focus Support in the unlikely event that you get this error.

ERROR: Failed to get SID for 'name', error=number (errortext)

(Windows only) This message, which only applies when file security is enabled, indicates that the system call to get the security ID for the given user name failed. Contact Micro Focus Support if you get this error. The Client will revert to using default security when this happens.

ERROR: Fetch of DATATABLES failed

This message can occur during a `define` or `redefine` command. It indicates that the FETCH of the external table names following the SQL select statement for the DATATABLES Client control table failed. For details, see the relational database API message that precedes this message (onscreen or in the log file).

ERROR: fgets failed for console input, errno=number (errortext)

(UNIX only) The dbutility console reads the keyboard to get a console command. If you run dbutility as a background run, an attempt to read from the keyboard will result in this error and cause the console thread to terminate. On UNIX, you can use a couple of the kill signals to communicate with the background run.

ERROR: File "name" specified in -f option does not exist

This message, which is confined to dbutility, indicates the argument of the `-f` option is not a valid filename. If you just specify a file name, the program looks for the file in the `config` directory. If you want to point to a file in any other place you need to specify the fully qualified name of the file.

ERROR: fseek failed for Null Record file, errno=number (errortext)

This error can occur when reading or write the NULL record files. If the error occurs during a process or clone when trying to read the file, you can try running a `redefine` command with a `-R` option to rebuild the file. If you get this error during a `define` or a `redefine` command, the only recourse you have is to set the parameter `read_null_records` to `False`.

ERROR: func masking function parameter contains a decimal point for item 'name' in table 'name' data type = dtype

This error message, which is limited to the SQL Server client, indicates that the validation of the masking parameter string specified in the configuration file failed for the column in question. This could happen for a masking type of "random" applied to a column that has in integer data type or a masking type of "partial" that has 3 argument, two of which are integers.

ERROR: func masking function parameter contains non-numeric characters for item 'name' in table 'name' data type = dtype

This error message, which is limited to the SQL Server client, indicates the validation of the masking parameter string specified in the configuration file failed for the column in question. This is limited to a masking type of "random" which has two numeric parameters.

ERROR: Generation of AA item 'name' in DATAITEMS table failed for 'tabname'

This message can occur during a `define` or `redefine` command. It indicates that the specified item could not be placed in the `DATAITEMS` table. The `AA Value` is the offset of the record in the `DMSII` data set (that is, its absolute address). The column, which is named `my_aa` or `my_rsn`, contains the `AA Value` or `RSN` (record serial number) of the record. For more information about the cause of the failure, see the relational database message that precedes this error.

ERROR: Generation of BITOR function failed

The Oracle client uses a `BITOR` function to perform logical OR operations on the various options column in the control tables. The `generate` command creates this function when it does not exist. Oracle has a native `BITAND` function that performs logical AND operations. This message indicates that the creation of the `BITOR` function failed. For more information about the cause of the failure, see the relational database messages that precede this error.

ERROR: Generation of Client Control Table version entry in DATASOURCES failed

This message can occur during a `dbutility configure` command or when the control tables are created by the Client Configurator. It indicates that the Client cannot generate the `DATASOURCES` table entry that holds the version of the Client control tables. For details about this problem, see the relational database messages that precede this error. One possible cause is that the `userid` does not have the required privileges.

ERROR: Generation of common scripts failed

This message applies to the Databridge Client for Oracle only. It can occur during a `generate` command, and it indicates that an error occurred while running one of the three common scripts that create or replace stored procedures used by the scripts the program uses to create or drop tables and stored procedures. The stored procedures are named `exec_DDL`, `drop_proc`, and `drop_table`. Try dropping these procedures using `SQL*Plus` and then rerun the `generate` command. The OCI messages that precede this error should provide further information as to why this error occurred. The most likely cause for this error is that you do not have the appropriate privileges to perform this operation.

ERROR: Generation of DATASETS entry failed for DataSet *name[/rectype]*

This message can occur during the `define` or `redefine` command. It indicates that an error occurred while inserting an entry into the Client control table `DATASETS` for the specified data set `name`. For more information about the cause of the failure, see the relational database message that precedes this message (onscreen or in the log file). See also [“OCI Errors” on page 157](#) or [“ODBC Errors” on page 156](#).

ERROR: Generation of DATASETS entry failed for Global\_DataSet

This message, which can occur during a `define` or `redefine` command. It indicates that an error occurred while inserting an entry for `Global_DataSet` into the `DATASETS` Client control table. This entry is used for holding the global stateinfo during update processing when the `optimize_stateinfo` and/or the `use_globalstateinfo` configuration parameters are enabled. For more information about the cause of the failure, see the relational database message that precedes this message (onscreen or in the log file). See also [“OCI Errors” on page 157](#) or [“ODBC Errors” on page 156](#).

ERROR: Generation of DATASOURCES entry failed for *name*

This message can occur during a `define` or a `redefine` command. It indicates that the Client could not insert a record into the `DATASOURCES` Client control table. For more information about the cause of the failure, see the relational database message that precedes this message (onscreen or in the log file). See also [“OCI Errors” on page 157](#) or [“ODBC Errors” on page 156](#).

ERROR: Generation of DATATABLES entry failed for '*name*'

This message can occur during the `define` or `redefine` command. It indicates that the Client could not insert a record describing the specified table into the `DATATABLES` table. For more information about the cause of the failure, see the relational database message that precedes this message (onscreen or in the log file). See also [“OCI Errors” on page 157](#) or [“ODBC Errors” on page 156](#).

ERROR: Generation of external column item '*name*' in `DATAITEMS` table failed for '*tabname*'

This message, which can occur during a `define` or a `redefine` command, indicates that the Client could not generate the entry for the external column item in the `DATAITEMS` Client control table. For details, see the relational database message that precedes this message (onscreen or in the log file).

ERROR: Generation of item *name* in `DMS_ITEMS` table failed for DataSet *name*[/*rectype*]

This message can occur during the `define` or `redefine` command. It indicates that an error occurred while inserting an entry into the Client control table `DMS_ITEMS` for the specified item and data set. See the relational database API message that precedes this message (on the screen or in the log file) for more information. See also [“OCI Errors” on page 157](#) or [“ODBC Errors” on page 156](#).

ERROR: Generation of [KEY] item '*name*' in `DATAITEMS` table failed for '*tabname*'

This message, which can occur during a `define` or a `redefine` command, indicates that the Client could not insert an entry into the `DATAITEMS` Client control table for the specified item, which is a key in the index for the specified table. See the relational database API message that precedes this message (on the screen or in the log file) for more information.

See also [“OCI Errors” on page 157](#) or [“ODBC Errors” on page 156](#).

ERROR: `getcwd` failed, `errno=number` (*errortext*)

See message "ERROR: `chdir` failed for directory "*path*", `errno=number` (*errortext*)", as these errors happen under very similar conditions.

ERROR: History table definition failed for DataSet *name*[/*rectype*]

The `define` or `redefine` command could not create the entry for the history table for the specified data set in the `DATATABLES`, or it could not create the entries for its columns in the `DATAITEMS` Client control tables. See the relational database messages that precede this error for more information about what went wrong.

ERROR: History table '*name*' does not include a timestamp [or an identity] column

In order to use history tables, the tables must have a timestamp column to determine the order in which to apply the changes. In the case of the Databridge Client for Microsoft SQL Server, an identity column is equally well suited for this purpose. If the Client finds no such column during a `generate` command, it displays this error, which causes the `generate` command to fail. Make sure that you have not set `active=0` in `DATAITEMS` for the columns that the `define` command automatically creates for history tables.

ERROR: History table '*name*' does not include an '`update_type`' column

In order to use history tables, the table must have an update type column, which specifies the type of update involved (insert, delete, or update). If the Client finds no such column during a `generate` command, it displays this error, which causes the `generate` command to fail. Make sure that you have not set `active=0` in `DATAITEMS` for the columns that the `define` command automatically creates for history tables.

ERROR: Host password cannot be decoded

This message can occur when the Client reads a text configuration that contains an encoded password that is corrupted, most typically because the file was edited. If this is a text file, replace the password and use `dbpwenc` or the `export` command to encode it.

ERROR: Illegal argument for 'g' option

This option is used to pass information from the Client Manager Service (or daemon, on UNIX) to the spawned run. It isn't applicable to `dbutility`. Call Micro Focus Technical Support.

ERROR: Illegal concatenation for items 'name1' and 'name2', resulting column is too large

This error indicates that concatenation is illegal because the resulting column would exceed the maximum length for the corresponding data type.

ERROR: Illegal date value *val* for item 'name' in table 'name', day set to *newday* - Keys: *columnname* = *value*,...

This message only occurs when the `correct_bad_days` parameter is set to a non-zero value. It indicates that the DMSII date item contains invalid day values, which the Client is changing to make the date valid. A day value of 0 is changed to 1 with no warnings regardless of the value of the `correct_bad_days` parameter

ERROR: Illegal date value *val* for item 'name' in table 'name', month set to *newmonth* - Keys: *columnname* = *value*, ...

This message only occurs when the `correct_bad_days` parameter is set to 2. It indicates that the DMSII date item contains invalid month values, which the Client is changing to make the date valid.

ERROR: Illegal date value *val* for item 'name' in table 'name', {record will be discarded | date set to NULL} - Keys: *columnname* = *value*, ...

This message can occur during the `process` and `clone` commands. It indicates that the date extracted from the DMSII data is in error. The most likely causes of this error is bad DMSII data or an incorrect value in the `dms_subtype` column. This is not a fatal error. The date is stored as NULL or, if the item is a key, the record is discarded.

ERROR: Illegal date value *val* for item 'name' in table 'name', value of year out of range, {record will be discarded | date set to NULL} - Keys: *columnname* = *value*, ...

This message can occur during a `process` or a `clone` command. It indicates that the year portion of the date is invalid for the relational database data type. The year portion of an item with a SQL type of `smalldatetime` is limited to the range 1900–2079. Similarly, the year portion of an item of SQL type of `datetime` is limited to the range 1753–9999. The data types of `date` and `datetime2` that were introduced in SQL Server 2008 have a range of 0001-9999. If you are dealing with a SQL type of `smalldatetime` consider changing it to `datetime` instead.

To make the client use these data types (instead of `smalldatetime` and `datetime`) in SQL Server, set the parameters `use_date` and `use_datetime2` to True in the Client configuration file. No additional customization is needed.

The program recognizes a MISER date of 99999 as a special date used by MISER systems and stores it as 6/6/2079 when the SQL type is `smalldatetime` and 12/31/9999 when the SQL type is `datetime`. This is not a fatal error. The date is stored as NULL or, if the item is a key, the record is discarded.

ERROR: Illegal dms\_subtype number for item 'name' in table 'name', {record will be discarded | date set to NULL} - Keys: columnname = value, ...

This message indicates that a member of a DMSII date GROUP has a dms\_subtype value that is not 1, 2, 3, or 4. Note that the only acceptable DMSII GROUPs are those with 2 or 3 numeric items. DMSII date groups are set up by setting the DIOPT\_Clone\_as\_Date (2) option in the di\_options column of the DMS\_ITEMS entry for the group, followed by the setting of the dms\_subtype columns for the members of the group.

This message can also indicate that a DMSII item with a data type of REAL marked as to cloned as date, has an illegal dms\_subtype value. For a complete list of the valid dms\_subtype values for dates refer to the *Databridge Client Administrator's Guide*.

ERROR: Illegal hex character 'char' found in encoded string

This message can occur when the Client tries to decode passwords when reading the configuration file at the start of a run. Passwords are always encoded in binary configuration files. Password encoding is optional in text-based configuration files and can be done with the export command or the dbpwenc utility.

ERROR: Illegal month name mmm for item 'name' in table 'name', {record will be discarded | date set to NULL} - Keys: columnname = value, ...

This message indicates that the Client encountered an illegal month name while processing a DMSII ALPHA date. If the month names are not in English, you need to use the months specification in the configuration file. This is not a fatal error. The date is stored as NULL or, if the item is a key, the record is discarded after it is written to the corresponding file in the discards subdirectory.

ERROR: Illegal numeric data (value) for field name[number] in archive file

This error can occur during a reload command. If you did not modify the file, report the error to Micro Focus Technical Support.

ERROR: Illegal [numeric] time value number for item 'name' in table 'name', {record will be discarded | date set to NULL} - Keys: columnname = value, ...

This message indicates that the Client encountered an illegal numeric time value while processing the item which is being interpreted as a TIME(1). This is not a fatal error and is most likely caused by bad DMSII data. The date is stored as NULL or, if the item is a key, the record is discarded.

ERROR: Illegal numeric value specified for {count | length}

This error can occur during a tcptest command if an illegal numeric value is specified for the count or length parameter of the command. A value that causes the count to go negative also results in this error. Do not use extremely large values for the count, as this would make the test run for a very long time using up a lot of mainframe CPU time.

ERROR: Illegal numeric value specified for 'F' option argument

The argument of the /F option for the Client is an audit file number, which must be in the range of 1 to 9999.

ERROR: Illegal numeric value specified for port

This message can occur during the scanning of the port number from the command line argument for a `define` command. It indicates that the specified port number is not syntactically correct.

For example, “`dbutility define cin.aa.microfocus.com 12,5`” would generate this message because it is not a syntactically correct number. An example of correct syntax is the following:

```
dbutility define cin.aa.microfocus.com 1255
```

ERROR: Illegal numeric value specified for `-t` option argument

This message can occur during the scanning of the `-t` command-line option, which has a numeric argument. It indicates that the specified argument is not syntactically correct. You can specify the trace mask as a decimal number or a hexadecimal number which must be prefixed with “0x”.

ERROR: Illegal numeric value specified for 'V' option argument

The `-V` option is used with the `dbutility unload` command to specify the control table version. When you upgrade, the `dbfixup` program creates control tables that are readable by the client from which you are upgrading. This allows you to safely reload these control tables and use the previous version if you experience a problem.

ERROR: Illegal operator `nnn` in filter for table '`name`'

This message only occurs when there is an OCCURS table filter present for the given table. It indicates that the binary filter file "`dbfilter.cfg`" is malformed. Try recompiling the binary filter by using the `import` command of the **makefilter** utility to get a fresh copy of the binary filter. If the problem persists, contact Micro Focus Technical Support.

ERROR: Illegal string

This message can occur while processing text configuration files or the UNIX `globalprofile.ini` file. It indicates that an illegal string value was entered into the configuration file for a parameter whose argument is a quoted string. This message is always followed by a second error message that lists the input record.

In the case of Windows, the most common cause of this error involves the use of back slashes in file names. You must enter a back slash as two back slashes because the first is interpreted as a force character. Failure to do this results in this error.

ERROR: Illegal update type `dd` for table '`name`'

This internal message only occurs when the `aux_stmts` configuration parameter is set to a nonzero value. It can occur during the `process` or `clone` command, and indicates that the section of code that generates the SQL statement for an update encountered an undefined update type.

ERROR: Improper section header in configuration file line: `input_text`

This message, which can occur when a text configuration file is being processed, indicates that section header is not defined or is not formatted using the following syntax:

```
[SectionHeaderName]
```

ERROR: Improper test with NULL only legal operations are "=" and "!="



This message only occurs when there is an OCCURS table filter present for the given table. It indicates that the binary filter file "dbfilter.cfg" contains a bad test that has a second operand of NULL. You can only test for equality (=) or inequality (!= or <>) with NULL. Examine your filter source file, correct the error and try recompiling the binary filter by using the import command of the **makefilter** utility to get a fresh copy of the binary filter. If the problem persists, contact Micro Focus Technical Support. Under normal circumstances **makefilter** should be flagging the filter statement as being in error; this situation is not expected to ever happen.

ERROR: Incomplete script file "*name*", missing '/\*\*/'

This message can occur during a `process` or a `clone` operation, while running a script to create a data table. It indicates that the script file `script.create.tabname` was not created correctly or is corrupt. First, check that an error did not occur during the last `generate` command. If the script file is corrupt, run the `generate` command again to create a new script file. You will need to use the `-u` option to force it to generate new scripts.

ERROR: Index creation failed for control table *name*

This message can occur during a `dbutility configure` command or when the control tables are created by the Console or the Client Configurator. It indicates that an error occurred while creating the index for the specified Client control table. See the relational database message that precedes this message (on the screen or in the log file) for more information.

ERROR: Index creation failed for history table '*name*'

This message, which can occur during a `reorg` command, indicates that the command was unable to create the index for the given history table. See the database API error messages that precede this error message to determine why the creation of the index for this empty table failed.

ERROR: Index creation failed for table '*name*'

This message, which can occur during a `reorg` command, indicates that the command was unable to re-create the index for the given table. The `reorg` command typically will drop the index for a table before altering it, when a column that is a member of the index is involved in the `alter` command. The command also drops and recreates the index for table whose index type is changed (for example, a unique index is changed to a primary key). See the database API error messages that precede this error message to determine why the creation of the index for this empty table failed.

ERROR: Index thread failure, creating index directly

This message, which is limited to UNIX clients, indicates that the attempt to create the index thread failed. When this occurs, the main thread creates the index instead. If this error persists, refer to the thread creation error to determine why it is failing.

ERROR: Insufficient temporary value entries -- contact Micro Focus support

This message only occurs when there is an OCCURS table filter present for the given table. It indicates that the filter you are using is too complex for the program to handle. Contact Micro Focus Technical Support to get a new version of the **makefilter** utility that has a larger temporary value array. Alternatively you may want to consider simplifying your filter statement.

ERROR: Internal error, undefined column type *number* for *columnname* encountered in `Process_Archive_Record()`

This message indicates an internal error in the `reload` command. Contact Micro Focus Technical Support unless you modified the archive file.

ERROR: Invalid boolean argument specified for {*sched* | *verbose*} command

The command entered in the command-line console contains an invalid Boolean argument. A Boolean argument is of the form {yes | no} or {true | false}. This applies to the `sched` and `verbose` commands.

ERROR: Invalid concatenation for item '*colname*' in table '*tablename*'

The client only supports concatenation of two ALPHA items or two unsigned NUMBER items. You can also use a NUMBER item that is cloned as ALPHA in place of an ALPHA item, or an ALPHA item that is cloned as a NUMBER in place of a numeric item. Any other combination results in this error message being displayed by the `define` and `redefine` commands.

ERROR: Invalid database update type *number* received

This message can occur during a `process` or a `clone` command. It indicates that the Databridge host software returned an undefined update type (for example, an update type that is not CREATE, DELETE, MODIFY, STATE, MODIFY\_BI, MODIFY\_AI, LINK\_AI, DELETE\_ALL or DOC). To get more information on this message, you must get a trace of DBServer communications (-t 0x45) and send the trace to Micro Focus Technical Support.

**NOTE:** An alternative is to turn off cloning (set active column to 0) for the offending data set.

ERROR: Invalid length specification for datatype '*name*' in external column '*colname*'

This message, which occurs when text-based configuration files are being processed, indicates that the `sql_length` specification in the given `external_column` parameter is invalid for the corresponding data type.

ERROR: Invalid SQL type '*name*' for external column '*colname*'

This message, which occurs when text configuration files are being processed, indicates that the parameter `sql_type` in the given `external_column` specification is invalid.

ERROR: Invalid stop\_time or end\_stop\_time value *dddd*, *dddd*, values set to 0

This message can occur during a `process` or `clone` command if the `controlled_execution` configuration parameter is enabled. It indicates that the program detected an error in the values of the `stop_time` and `end_stop_time` columns it read from the DATASOURCES client control table. These values are integers that represent a time of a day using 24-hour time (*hh:mm* format). If the value *hh* is not in the range 0:24 and *mm* is not in the range 0:59, both entries are set to 0 and this error appears.

ERROR: Invalid structure index *number* received

This message indicates a problem on the host. In this case, the Databridge Engine is sending a DMSII structure index number that is invalid (for example, a structure index number less than 0). Structure indexes are assigned when the Client sends `DB_Select` requests to the Engine for each data set that is to receive updates or extracts at the start of a `process` or `clone` command. This index is used to associate DMSII records with the data set they belong to. A structure index of 0 is only used for STATE records, it indicates that the information applies to all selected data sets (that must all have a `ds_mode` of 2). To get more information on this message, you must trace DBServer communications (-t 0x45) and send the trace to Micro Focus Technical Support.

ERROR: Invalid value *nnn* for `dms_concat_num` column in DATAITEMS for table '*tablename*' item '*colname*'

This message indicates that the Client detected a bad item number in the `dms_concat_num` column. While this error can occur during any Client command, it typically happens when loading the control tables. The most likely cause of this error is from a non-existent item number in the `dms_concat_num` column. After a DMSII reorganization or when changes occur in GenFormat, DMS items numbers can change. If you use hard-coded numbers in your user scripts, you may end up concatenating different columns than the ones you originally specified. Use subqueries in your user scripts instead of hard-coded numbers.

Fix your user script and force a `redefine` command to make the change take effect. Set `status_bits` to 8 for the data set in question or use the `-R` option for the `redefine` command.

```
ERROR: IO errors in writing {bcp format | sqlldr control | script} file "name",  
      errno=number (errortext)
```

This message can occur during a `generate` or `createscripts` command. It indicates that an error occurred while writing to the specified file. The system error message `errortext` should give you a handle on why this error occurred.

```
ERROR: IO errors in writing file "name", errno=number (errortext)
```

This error message, which can occur during an `unload` command, indicates that an error occurred while writing a record to the archive file whose name appears in the message. The system error should explain the cause of the problem.

```
ERROR: It is illegal to change the keys for a DataSet when it is using a DMSII SET  
for the index
```

The Client Configurator currently does not support changing the values of `item_key` in `DMS_ITEMS` for data sets that are using a DMSII SET as the source for the index.

```
ERROR: Item 'name' (dms_item_type = number) cannot be cloned
```

This message, which typically should not occur, is an indication that the specified `dms_item_type` column contains a value that the program cannot deal with. If you happen to have a `DATAITEMS` entry whose type is `GROUP (29)`, attempting to set the active column to 1 results in this error, unless the `sql_type` of the item is set to `date`.

```
ERROR: Item 'name' in control table name has an illegal sql_type of nnn
```

This message, which is not likely to be seen, indicates that when attempting to set up the host variables for updating a control table a column with an invalid SQL data type code was encountered.

```
ERROR: Item 'name' in table 'name' cannot be cloned as three booleans
```

The program clones `DMSII NUMBER(1)` items as a field of three Booleans when the `DIOPT_Clone_as_Tribit` bit (16) in the `di_options` column of `DMS_ITEMS` is enabled. If you try to use this option with a `DMSII NUMBER` whose length is not 1 you get this error.

```
ERROR: Item 'name' in table 'name' cannot be cloned using a TIME data type,  
{record will be discarded | time set to NULL} - Keys: columnname = value, ...
```

This message occurs if you try to clone a `DMSII TIME(12)`, represented by a `dms_subtype` value of 4, as a SQL Server `TIME` data type. Use a numeric time instead, as this is not supported.

```
ERROR: Item 'name' in table 'name' cannot be flattened to a string result is too  
long
```

This message only occurs when you try to flatten a single item with an OCCURS clause to a string. It indicates that the resulting column is too long for a char or varchar data type. You need to consider other ways of dealing with this particular OCCURS clause.

ERROR: Item '*name*' (type *nnn*) cannot be tested for NULL as it is not nullable in DMSII

This message only occurs when there is an OCCURS table filter present for the given table. It indicates that the binary filter file "*dbfilter.cfg*" contains a test for NULL for an item that is not nullable in DMSII. You need to rewrite the filter statement for this table, as what you are trying will not yield the results expected.

ERROR: Item '*name*' in table '*name*' is not an unsigned NUMBER or an ALPHA item, flattening to a string is not supported

This message only occurs when you try to flatten a single item with an OCCURS clause to a string. It indicates that the item is not an unsigned NUMBER or an ALPHA item, which are the only two data types for which flattening to a string is supported.

ERROR: Item *name* in DataSet *name[/rectype]* points to a non-existent or inactive DataSet (strnum=*nnn*)

This message only applies when the configuration parameter *enable\_dms\_links* is enabled. It indicates that the specified item points to a non-existent or inactive data set. This situation is clearly an error, as the link must point to a valid table. You must either set the active column to 0 in the DMS\_ITEMS table for the link in question, or set the active column to 1 in the DATASETS table for the target data set of the link.

ERROR: Item '*name*' in table '*name*' contains an illegal numeric value *val*, {record will be discarded | date set to NULL | time set to NULL} - Keys: *columnname* = *value*, ...

This message indicates that the Client encountered an invalid number. The program recognizes numbers that have all their digits set to 0xF to be NULLS; any other number that has a value that is not 0–9 in any digit position except for the sign is treated as bad. A bad number is stored as NULL unless the item is a key, in which case the record is discarded. Note that if the *DAOPT\_Allow\_Nulls* bit (1) in the *da\_options* column of DATAITEMS is not set, the number is stored as all nines (9) or all zeros (0), depending on the setting of the configuration parameter *null\_digit\_value*.

ERROR: Item '*name*' in table '*name*' contains an invalid time value *hh:mm:ss*, {record will be discarded | time set to NULL} - Keys: *columnname* = *value*, ...

This message indicates that the Client detected an invalid time value while processing a DMSII item as a time value. This error can occur when the DMSII data is bad or when the item is not an actual DMSII time. For details on interpreting DMSII items as time values, see "DMSII Times" in Chapter 2 of the *Databridge Client Administrator's Guide*.

ERROR: Item '*name*' in table '*name*' has an illegal {*dms\_item\_type* | *dms\_subtype*} value of *nnn*

This message, which can occur during a *createscripts* command indicates that the item in question is in error. This is the result of bad user scripts. You need to fix this before proceeding any further, as it will cause the Client to fail.

ERROR: Item '*name*' in table '*name*' points to non-existent DMS item numbered *nnn*

This message, which can occur during a `createscripts` command indicates that the item in question is in error. This is the result of bad user scripts. The most likely cause of this error is that the item number in the `dms_concat_num` column does not exist. You need to fix this before proceeding any further, as it will cause the client to fail.

```
ERROR: Item 'name' in table 'name' which is a member of a date group is not an integer value, {record will be discarded | date set to NULL} - Keys: columnname = value, ...
```

This message appears when you have defined `sql_type` to a date data type and `dms_subtype` to 1, 2, 3, or 4, but there is a non-numeric member in the DMSII GROUP. The DMSII date GROUP can contain only numeric fields that must be appropriately identified as year, month, and day. For more information on setting the `dms_subtype` column for dates, see Chapter 2 in the *Databridge Client Administrator's Guide*.

```
ERROR: Last Database Error = number
```

This message is printed when a SQL error occurs while processing updates. This message is followed by the actual SQL statement that provoked the error.

```
ERROR: Load of Databridge control tables failed
```

This message can occur during any Client command except for `dbutility configure`, `refresh`, `runscript`, and `tcptest`. It indicates that an error occurred while reading the client control tables. See the relational database API message that precedes this message (on the screen or in the log file) for more information. See also [“OCI Errors” on page 157](#) or [“ODBC Errors” on page 156](#).

```
ERROR: Log file prefix is too long, maximum allowable length is 20, value truncated
```

You can specify a prefix for the Client log files in the Client configuration file, however this prefix is limited to 20 characters. If you use a longer prefix, you get this error. The default prefix is “db”. We recommend using the data source name as the prefix, when you have more than one data source.

```
ERROR: Logswitch command failed
```

The log switch command, which can be issued from the Console, closes the current log file and open a new one with a different name. See the error messages that precede this message in the log file to see why the command failed.

```
ERROR: Mask type value nn out of range
```

(SQL Server Client only) This message indicates that the masking type, which resides in the low 8-bits of the `masking_info` column of the `DATAITEMS` table has a value that is not in the range 0 to 4. It is an indication that you have a bad user script.

```
ERROR: Masking function 'func' does not support parameters, masking string ignored
```

(SQL Server Client only) This message indicates the masking type, which resides in the low 8-bits of the `masking_info` column of the `DATAITEMS` table does not have parameters. This includes the “default” and “email” masking functions. Zero the rest of the entry to fix this problem, which is caused by a bad user script.

```
ERROR: Maximum bcp errors threshold exceeded for table 'tablename', load aborted
```

(SQL Server Client only) This message indicates that the table in question has gotten more discards than the value specified while issuing `bcp_sendrow` calls into the BCP API. At this point the client will close off the bcp connection and silently discard any additional records received for this table, which will end up needing to be reloaded.

ERROR: Merge of neighboring items only valid for ALPHA and unsigned NUMBER - unable to merge items 'name1' & 'name2'

The Client merges two neighboring items of like type to form a bigger item when the `di_options` bit `DIOPT_MergeNeighbors` (0x1000000) is set in the `DMS_ITEMS` table entry for the first item. This feature is only supported for items of type ALPHA or unsigned NUMBER. If you try to merge any other type of items you get this error.

ERROR: Mismatched AFN values for reorganized DataSets: 'name1' AFN = *afn1* and 'name2' AFN = *afn2*

This message, which can occur during a `redefine` command, indicates that not all the data sets to be reorganized have the same AFN value in their `stateinfo`. Most likely, the value of the active column changed for one of the data sets.

ERROR: Missing END operator in filter for table 'name'

This message only occurs when there is an OCCURS table filter present for the given table. It indicates that the binary filter file "`dbfilter.cfg`" contains a filter that does not end in an END operator. This indicates that an internal error occurred in the `makefilter` utility that should never happen; contact Micro Focus Technical Support.

ERROR: Missing entry points in DLL "name"

(SQL Server Client only) When using the BCP API, the Databridge Client loads the ODBC DLL and sets up a table of addresses through which it makes the BCP API calls. This allows the Client to work with the version of the ODBC DLL that supports the features needed. Microsoft includes the SQL Server version number in the DLL name, so we can link the DLL, as its name changes with every version of the ODBC driver.

This message indicates that one of the BCP API entry points that we need is not present in the ODBC driver being used.

ERROR: Missing length specification for SQL type *sql\_type* (*nn*) in external column '*colname*'

This error, which can occur during text configuration file processing, indicates that the `sql_length` specification for the given `external_column` parameter is invalid.

ERROR: *name* command requires a {boolean | decimal numeric | valid | valid numeric | valid string} argument

This message is a response to bad input from the `dbutility` command-line console for a command, which requires the given type of argument. Boolean arguments can be True or False (or "T" or "F"). Decimal arguments cannot be entered as hexadecimal values. Numeric values can be decimal or hexadecimal numbers. Hexadecimal numbers must be prefixed by "0x". String arguments are typically text and optionally enclosed in double quotation marks. You must use double quotation marks when the data contains a non-alphanumeric character.

ERROR: Name 'name' is not a valid {user | group} name

This message, which only applies when file security is enabled, indicates that the system call to get the security ID for `userid` or a group name failed. The client uses this call to determine if the user is allowed to run the client and when setting the ACL for a file or directory it creates. If you need to change the file security settings for the client, use the “`setfilesecurity`” program for this. Do not use `regedit`.

ERROR: No active structures were found

This message can occur during a `process` or `clone` command. It indicates that no data sets are selected for cloning or updating. In other words, the active columns in the DATASETS client control table are set to 0 (cloning off). This situation could occur when you use a SQL statement to change the value for the active column, but you do not use a WHERE clause.

ERROR: No configuration file *name* specified as an argument to the 'f' option

This message indicates that the `-f` option is not followed by a file specification. You cannot specify a null configuration file by omitting the parameter for the `-f` option.

ERROR: No data received from {DBServer | DBEnterprise Server} for *nnn* minutes, aborting client

This message is displayed when the parameter `max_srv_idle_time` is set to a non-zero value. It indicates that no data was received from the server for the specified amount of time and the client is about to stop. The client will exit with an exit code of 2059 after resetting the connection to the server. When using the service, the service will attempt to restart the client after a brief delay.

ERROR: No item with `dms_subtype` set to 254 was found in table '*name*'

This message will only occur when you replicate DMSII embedded subsets using virtual data sets. It indicates that the virtual data set is not properly defined. A value of 254 in the `dms_subtype` column of the item is used to indicate that the item is a parent key (that is, it contains the AA Value of the parent item). The virtual data set in question thus implements the embedded subset.

ERROR: Non-link item '*name*' in table '*tablename*' cannot follow link items, command aborted

When using DMSII links, the Client requires that all links be placed at the end of the table. Adding non-DMSII columns to the end of the table results in this error. To rectify this problem, you need to change the value of the `item_number` columns of the links in DATAITEMS so that they land after all non-link items.

ERROR: Null Record file does not contain an entry for DataSet *name*[/*rectype*]

This message, which can occur during a `process` or `clone` command, indicates that the null record entry for the specified data set is missing from the file. The most common cause of this error is enabling the configuration parameter `read_null_records` for a data source that was already replicated. To rectify this problem, run a `redefine` command with the `-R` option to rebuild the null record file.

ERROR: Null Record file "*datasource\_NullRec.dat*" is corrupt

This message, which can occur during the `define` and `redefine` commands, indicates that the specified file is corrupt because the Client could not locate a record that is supposed to be in the file. To rectify this problem, re-run the `redefine` command with the `-R` option to rebuild the null record file.

**NOTE:** If you reload the Client control tables from the unload file that the command creates and rerun the command, you will most likely get this message, which you can safely ignore.

ERROR: Old masking string is not applicable to masking function '*func*'

(SQL Server Client only) When you use the Client Configurator to setup the masking string, it manages the string in conjunction with the client (DBCIntCfgServer). If you change the masking type and keep the old masking string index, you will end up getting this message. To fix this, replace the masking string or make it be empty. The client will remove unused strings and it will reuse entries that are already present.

ERROR: Open failed for archive file "*name*", *errno=number (errortext)*

This error, which can occur during a Client `reload` command or a DBCIntCfgServer `configure` command that drives the Client Configurator. It indicates that the program got a system error when trying to open the archive file.

ERROR: Open failed for file "*name*", *errno=number (errortext)*

This message can occur during any command that attempts to open a new file for write. The included system error should explain the cause of the problem. In order to implement file security on Windows, all such file opens use common code that creates the file using the Windows library `CreateFile` procedure, which allows a DACL that defines the file security to be supplied. Once the file is created with the proper security we close it and reopen using ANSI C library procedures, you will get this error in the unlikely situation where the `fopen` or `open` call fails. In the case of UNIX clients all such file open operations go through common code that displays this message in case of error.

ERROR: Open failed for filter file "*name*", *errno=number (errortext)*

This message only occurs when there is an OCCURS table filter file "`dbfilter.cfg`" present in the config subdirectory for the data source. It indicates that the attempt to open the filter file for read failed. The provided system error number and its associated test string should provide you with some clues about why this error occurred. You can try deleting the binary filter file and rerunning the `makefilter import` command to recreate the file.

ERROR: Open failed for global configuration file "`/etc/Micro Focus/Databridge/6.6/globalprofile.ini`"

In order to be able to run UNIX clients you must first create the file `/etc/Attachmate/DATABridge/6.6/globalprofile.ini`. This file serves the same purpose as the Windows registry for the Client. It defines the directories where the software was installed and the global working directory where the client lock files are created in the locks subdirectory. This file also defines the userid under which the daemon is to be run. If you do not create this file, the Client will display this error and exit.

ERROR: Open failed for Null Record file "*name*" failed, *errno=number (errortext)*

This message indicates that the Client failed to open the file `datasource_NullRec.dat`. The most common cause of this error is enabling the configuration parameter `read_null_records` for a data source that was already replicated. To rectify this problem, run a `redefine` command with the `-R` option to rebuild the null record file.

ERROR: Open failed for pipe for shell "*filename*", *errno=number (errortext)*

This message, which applies to UNIX, can occur during a `process` or `clone` command. It indicates that the program could not open the shell script file which launches the SQL\*Loader. The included system error should explain the cause of the problem.

ERROR: Open failed for script file "*name*", *errno=number (errortext)*



Example: "ERROR: Open failed for script file "*name*", errno=2, (No such file or directory)"

This message can occur during a `process`, `clone`, `generate`, `refresh`, `runscript` or `createscripts` command. It indicates that the Client cannot find or open the specified script file. This error typically occurs if the `dbscripts` sub-directory does not contain the scripts or the `user_scripts_dir` parameter is not properly set up. Make sure that your scripts are in the directories where they are expected to be (`dbscripts` or `scripts`) for this data source and that you have not inadvertently deleted any script files. If you ran a `createscripts` command before the error occurred, the `user_script_dir` parameter may point to a nonexistent directory.

ERROR: Open\_stmt failed for thread[*nn*]

This error, which is limited to multi-threaded updates, is usually caused by an SQL error while creating the STMT. An STMT is a data structure that is used to execute pre-parsed SQL statements using host variables to pass the data. It could also be the result of memory allocation error. See the preceding error message for details about the problem.

ERROR: Operand stack {overflow | underflow}-- contact Micro Focus support

This message only occurs when there is an OCCURS table filter file "`dbfilter.cfg`" present in the config subdirectory for the data source. It indicates that an internal error has occurred while executing the filter pseudo-code. Try to recreate the filter using the **makefilter** utility and if that fails contact Micro Focus Technical Support.

ERROR: Operands for logical operator {AND | OR} are not both boolean values

This message is an internal error that only occurs when there is an OCCURS table filter file "`dbfilter.cfg`" present in the config subdirectory for the data source. It indicates that the operands for an "AND" or "OR" operation are not both boolean values. Examine the filter source file "`dbfilter.txt`" to see if you can see something wrong with the filter statement for the given table. If you can see the problem, fix the statement and recompile the filter using the **makefilter** utility. If that fails or you cannot see anything wrong with the filter command, contact Micro Focus Technical Support.

ERROR: Parent DataSet for table number *nnn* referenced in filter does not have the `ds_options "DSOPT_FilteredTable"` bit set  
- Run the `makefilter` utility to create a new filter file that is current

This message is an internal error that only occurs when there is an OCCURS table filter file "`dbfilter.cfg`" present in the config subdirectory for the data source. It indicates the binary filter and the Client control tables are not in sync. Try recreating the filter by either running the **makefilter** utility's `import` command or run a `redefine` command with the `-R` option (Redefine All) to try to remedy this situation. This situation should never occur, as whenever you run a `redefine` command or you run the Client Configurator **makefilter** is automatically launched. However, you need to pay attention to the cases where the program is unable to compile the filter due to errors in the filter's source file. Look in the **makefilter** log file to determine why the compile failed.

ERROR: Parser table in error

This is an internal error indicating that the parser table for the Windows console command is in error. Contact Micro Focus Customer Care if you get this error.

ERROR: Partial load failed for table '*name*' using `sql*loader` - see file "`sqlld.tablename.log`" for more information

This message applies to Oracle (Windows). It can occur when SQL\*Loader stops processing during a segmented load of a table. The amount of the segmented load is determined by your setting for `max_temp_storage`.

ERROR: partial() masking function for item 'name' in table 'name' data type = *dtype* requires three parameters

(SQL Server Client only) When you specify a partial masking function for a column it must have three comma separated parameters; a numeric prefix, a text padding, and a numeric suffix. The prefix and the suffix specify the number of characters to expose at the start and the end of the string, while the padding replaces the remaining characters. This error indicates that the masking string you created does meet these specifications.

ERROR: partial() masking function is illegal for item 'name' in table 'name' data type = *dtype*

(SQL Server Client only) This error indicates that you are trying to use the partial masking function for a column whose data type is not char or varchar.

ERROR: Pass1 of two pass modify failed (return code = *ddd*)

This message, which can occur during a `process` or `clone` command, indicates that an error occurred while processing an update for an item with an OCCURS DEPENDING ON clause. The value of the item pointed to by the OCCURS DEPENDING ON changed. The Client updates the rows that remain in the table and then inserts new rows or deletes rows that no longer exist from the OCCURS table. The following values indicate the status of updating the tables, and the Client handles them appropriately:

- ◆ 6 - rows to insert
- ◆ 7 - rows to delete
- ◆ 8 - one of the updates found no matching rows

If any other value is returned, it is probably an internal error or some other error that caused the Client to return an unexpected status. Contact Micro Focus Technical Support if this error is not caused by another error during the process of updating the table.

ERROR: Prepare failed for SQL statement: *sql\_stmt*

This message can occur during a `process` or `clone` command. It indicates that an error occurred while parsing a SQL statement used to call stored procedures for updating data or control tables. See the relational database API message that precedes this message (on the screen or in the log file) for more information. See also [“OCI Errors” on page 157](#) or [“ODBC Errors” on page 156](#).

ERROR: Processing of configuration file "*name*" failed

This message occurs any time the Client finds an error in the text-based configuration file that causes it to terminate.

ERROR: Program terminating, due to bulk loader failure

This message can occur during a `process` or `clone` command. It indicates that there was a bulk loader failure during data extraction and that the Client is terminating. When you set the `verify_bulk_load` parameter to True, a bulk loader error causes the client to abort the clone. This avoids having to extract all the data for a data set, only to find out that the bulk loader count verification failed, which causes the client to eventually abort.

ERROR: Program terminating, error occurred in a worker thread

This message indicates that a fatal error occurred in an Update Worker thread causing the program to terminate. This message is displayed by the main thread once it realizes that such an error has occurred. To determine what's causing the problem in the Update Worker thread, look for the additional error messages in the log file.

ERROR: pthread\_create failed for 'name', error=number (errortext)

(UNIX) This error indicates that the system could not create the specified thread. The Index Creator Thread creates indexes for tables whose data extraction completes successfully. The Watchdog Thread performs periodic checks for things such as lack of response from the server. When using multi-threaded updates, several Update Worker Threads are responsible for executing SQL.

This is an internal error that should never occur. It is an indication that the system might be low on resources.

ERROR: pthread\_mutex\_init failed for 'name', errno=error (errortext)

(UNIX) This error indicates that the initialization of a mutex in question failed. It is an internal error that should never occur.

ERROR: QueryPerformanceCounter failed, error=number (errortext)

(Windows only) This is an internal error that indicates that the client was unable to get information about the granularity of the external clock that we use to collect time statistics. Contact Micro Focus support if you get this error. The only time you could legitimately get this error is if you are using ancient hardware that does not have an external clock.

ERROR: quit after command requires an audit file number in the range 0 to 9999

This message can occur during a process or clone command and indicates that the AFN for the QUIT command is invalid.

ERROR: quit command not in the form: "QUIT {AT hh:mm | AFTER nnnn}"

This message indicates that there is a syntax error in the dbutility console QUIT command issued by the operator.

ERROR: random() masking function is illegal for item 'name' in table 'name' data type = dtype

(SQL Server Client only) This error indicates that you are trying to use the random masking function for a column whose data type is not a numeric value (for example int or decimal).

ERROR: Read failed for file "name", errno=number (errortext)

This message, which only displays on the screen, indicates that the log descriptor file log.cfg could not be read. This file is a tiny binary file used by the client to keep track of the log file name. If the error persists simply delete this file and let the client create a new one. The most likely source of this error is file ownership conflicts between the command-line client and the service. See the system error in this message for more information about why this error occurred.

ERROR: Read failed for Null Record file, errno=number (errortext)

This message, which can occur during a process, clone, or redefine command, indicates that an I/O error occurred while reading the null record file. To rectify this problem, try running a redefine command with the -R option to rebuild the null record file.

ERROR: Read failed for trace descriptor file "name", errno=number (errortext)

The trace descriptor file `trace.cfg` is a tiny binary file used by the client to keep track of the trace file name. If the error persists simply delete this file and let the client create a new one. The most likely source of this error is file ownership conflicts between the command-line client and the service. See the system error in this message for more information about why this error occurred.

ERROR: ReadFile failed for console, *error = number*

This message indicates that the Console thread for the Windows Client received a read error while reading keyboard input. You typically get this error message when the Client is terminated by pressing Ctrl+C.

ERROR: Real data set link for data set *name[/rectype]* is NULL, make sure that `automate_virtuals` is true

In a MISER database, user scripts create a pointer that links the virtual data sets and the real data sets from which they are derived using the `virtual_ds_num`, `real_ds_num` and `real_ds_rectype` columns in the DATASETS Client control table. If the configuration parameter `automate_virtuals` is not enabled, this pointer is not set up and executing a `createscripts` command returns this message.

ERROR: Received invalid structure index for DataSet *name[/rectype]*

This message can occur during a `process` or `clone` command. It indicates that DBServer returned a negative structure index in the response packet for a `DB_Select` RPC call. When this occurs, set the active column to 0 in the DATASETS table for the specified data set and try again. If this error continues, contact Micro Focus Technical Support.

ERROR: Refresh command failed

This message indicates that `dbfixup` set a bit in the data source's `status_bits` column to indicate that there are OCCURS tables present. Upon seeing this bit a `process` or `clone` command initiates a `refresh` command to get the stored procedures `z_tablename` created. These stored procedures are used to speed up delete operations for such tables. Rather than deleting the rows of secondary table for a given key one by one, we delete them all in a single SQL statement using this stored procedure. This message indicates that the launched command failed, look at the client log file for clues as to why the command failed.

ERROR: Refresh of stored procedures failed for DataSet *name[/rectype]*

This message, which can occur during a `reorg` command, indicates that the Client was unable to drop and recreate the three stored update procedures that are associated with the tables for the data set. See the preceding SQL error to figure out what causes the error.

ERROR: ReleaseSemaphore failed for '*name*', *error=number (errortext)*

This message, which can occur during a `process` or `clone` command for Windows Clients, indicates that an error occurred while attempting to post the semaphore (either the `bcp_work_semaphore` or the `index_work_semaphore`, which pass work items to the corresponding threads during the data extraction phase). This is a system error which should not occur under normal circumstances. Reboot Windows.

ERROR: Resequencing DATAITEMS table entries failed for '*itemname*' of table '*tablename*'

The `define` and `redefine` commands resequence DMSII links to always appear at the end of data tables. This happens because, during data extraction, data for the links is received as a separate record and must be added to the previous record that contained the data part of the data set record. This message indicates that a SQL error occurred during the resequencing. See the relational database API message that precedes this message (onscreen or in the log file) for more information.

ERROR: RPC response length of `dddddd (0xhhhhhhh)` is too large

This message, which is very unlikely to occur, indicates that an RPC response packet has a bad length word. All RPC responses are preceded by a 4-byte length. This error indicates that the message is too long to be valid. In the unlikely event that you get this error, simply restart the client. If you still get the error, call Micro Focus Support.

ERROR: Script generation failed

This message can occur during a `generate` command. It indicates that the scripts could not be generated. Typically, this message is preceded by other more explanatory messages.

ERROR: Scripts for DataSet `name[/rectype]` are not current; you must first run a `generate` command

This message, which can occur at the start of a `process` or `clone` command, indicates that the program believes that you need to run a `generate` command. The `DS_Needs_Generating` bit of the `status_bits` column of the DATASETS entry is used to keep track of this. As stated, run a `generate` command before going any further.

ERROR: Scripts must reside within the global Databridge Client working directory

This error, which can only occur when file security is enabled, indicates that the user scripts are not in a subdirectory of the client working directory. You need to place your scripts either in the `scripts` subdirectory of the data source's working directory or in a subdirectory of the Client working directory when these scripts are shared among various data sources (in this case you could name this directory "userscripts").

ERROR: Select of DATATABLES failed

This message can occur during a `define` or `redefine` command. It indicates that the SQL `SELECT` statement used to get the external table names from the client control table `DATATABLES` failed. See the relational database API message that precedes this message (on the screen or in the log file) for more information. See also ["OCI Errors" on page 157](#) or ["ODBC Errors" on page 156](#).

ERROR: `sem_init` failed for '`name`', error=`number` (`errortext`)

(UNIX) This error indicates that the initialization of a semaphore failed. The Client uses several semaphores to synchronize activities between the various threads. This is an internal error that should never occur, unless the system is low on resources.

ERROR: `Send_DS_Added_Msg()` failed

This is an internal error that indicates that the Client was unable to send an IPC message to the service for forwarding to the Console. The most common cause of the error would be a network error.

ERROR: `Send_DS_Deleted_Msg()` failed

This is an internal error that indicates that the Client was unable to send an IPC message to the service for forwarding to the Console. The most common cause of the error would be a network error.

ERROR: Send\_DS\_Mode\_Chg\_Msg() failed

This is an internal error that indicates that the Client was unable to send an IPC message to the service for forwarding to the Console. The most common cause of the error would be a network error.

ERROR: Send\_IPC\_Message() failed

This is an internal error that indicates that the Client was unable to send an IPC message to the service for forwarding to the Console. The most common cause of the error would be a network error.

ERROR: Send\_IPC\_Response() failed

This is an internal error that indicates that the Client was unable to send an IPC message to the service for forwarding to the Console. The most common cause of the error would be a network error.

ERROR: Set of DATEFORMAT failed

This message applies to the Client for Microsoft SQL Server. It indicates that the attempt to override the database servers default date format was not successful. See the ODBC message that precedes this message (onscreen or in the log file) for more information.

ERROR: Set of nocount off failed

This message applies to the Client for Microsoft SQL Server. It can occur when you first start the Client and it indicates that the attempt to enable row counts by execution the SQL statement `SET NOCOUNT OFF` failed. See the ODBC message that precedes this message (on the screen or in the log file) for more information. See also ["ODBC Errors" on page 156](#).

ERROR: SetEntriesInAcl() returned number (errortext)

(Windows only) This message can occur when the Client tries to create an ACL that is uses to create a file or a directory. It indicates that the Windows procedure, which converts of an array of security entries into an ACL, failed. The accompanying error number and error text should help determine what is causing this problem. The most likely source of this error is that the user id under which the Client is running does not have the proper permissions to be able to create an ACL. You may want to temporarily revert to using the default security, until you get this problem resolved.

ERROR: SHCreateDirectoryEx failed for file "name", errno=number (errortext)

(Windows only) This message can occur when the command-line client tries to create the working directory and the operation fails. The accompanying error number and error text should help determine what is causing this problem. The most likely source of this error is that the user id under which dbutility is running does not have the proper permissions to be able to create the working directory.

ERROR: Source record missing in unload file "name"

This message, which can occur during a `DBCIntCfgServer configure` command (which is not the same as a `dbutility configure` command), indicates that the unload file that is being used to hold the backup copy of the client control tables does not have a source record (S, ...) immediately following the version record (V, ...). It is an indication that the unload file is corrupt. If you haven't modified this file, contact Micro Focus Technical support.

ERROR: sp\_recompile failed for table 'name'

This message applies to Microsoft SQL Server. It can occur during a `process` or `clone` command after an index for a table is created. The `sp_recompile` stored procedure informs the relational database that all the procedures associated with the table should be recompiled at the next execution. This ensures that the query plans associated with the tables use the index that was just created.

ERROR: SQL operation [for table '*name*'] timed out (elapsed time *eee*, incident # *nn*), aborting query

This message is returned by the watchdog timer thread when the client wait time for a SQL operation to complete reaches the secondary threshold specified of the parameter `sql_exec_timeout`. (A value of 0 disables this timeout.)

If the table name is known to the client, it is included in the message. The value *eee* is expressed in the appropriate units based on its value (e.g., 15 minutes). The value *nn* is a thread specific counter that is used to make it easy to see if this is a new problem or a repeat warning for an existing hang. When this situation occurs the client stops with an exit code of 2058.

ERROR: SQLAllocHandle(SQL\_HANDLE\_ENV) failed

This error, which can occur with any ODBC or CLI clients, indicates that the ODBC `SQLAllocHandle` call for the environment handle failed. Check the preceding ODBC error message for more information about the reason for the failure. This error should only occur if the system is totally out of memory.

ERROR: SQLGetDiagRec returned *error\_name*

This message, which can occur in ODBC and CLI clients, indicates that an error occurred while attempting to retrieve an ODBC error. The string *error\_name* is one of the following: "SQL\_INVALID\_HANDLE", "SQL\_STILL\_EXECUTING", "SQL\_NEED\_DATA", "SQL\_ERROR", "SQL\_SUCCESS\_WITH\_INFO", "SQL\_NO\_DATA\_FOUND" or "Unknown Error(*nnn*)". Contact Micro Focus Technical Support if you get this error.

ERROR: SQL\*Loader control file entry for item '*name*' (dms\_item\_type = *tt*) cannot be generated

This message, which applies to the Client for Oracle, can occur during the `generate` command when creating the SQL\*Loader control file. If an item whose `dms_item_type` column contains an illegal value is encountered, the program displays this error. This message originates from exactly the same conditions as the message "ERROR: Item '*name*' (dms\_item\_type = *number*) cannot be cloned," which can occur during a `process` or `clone` command.

ERROR: Stmt allocation failed for table '*name*'

This message can occur during a `process` or `clone` command, and it indicates that the Client was unable to create a STMT for processing an update to the specified table. See the database API error that precedes this message to determine the cause. The parameter `aux_stmts` may be set too high for your hardware configuration. Try reducing the number to 25. This is most likely an indication that you do not have enough memory on your system, or that there is a memory leak. See also "[OCI Errors](#)" on page 157 or "[ODBC Errors](#)" on page 156.

ERROR: system command failed for file "*name*", errno=*number* (*errortext*)

This message, which only applies to Windows Clients, can occur during a `process` or `clone` command. It indicates that an error occurred while spawning a command prompt session to run the bulk loader utility (SQL\*Loader for Oracle and `bcp` for Microsoft SQL Server).

ERROR: Table definition failed for DataSet *name*[/*rectype*]

This message can occur when you run `define` or `redefine`. It indicates that the client control tables were not populated. For details on why this occurred, see the error messages that occurred during processing of the data set specified by *name*.

ERROR: Table '*name*' does not contain a `res_flag` column

This message, which only applies to MISER databases, indicates that a multi-input virtual data set does not have the column it requires for tracking resident history records. Make sure that you did not inadvertently set `active` to 0 for this column; its name usually contains the text "`res_flag`" and its `dms_subtype` value is 255.

ERROR: Table name prefix for DataSource *name* is too long

This message is an internal error, which indicates that the `tab_name_prefix` value, read from the DATASOURCES client control table, is longer than 8 characters. The only way this can happen is if you alter the DATASOURCES table and increase the length of this column.

ERROR: Table '*name*' which has an index defined has no key items, index script generation failed

This error occurs if you disable cloning for all the key fields in the index for the specified table. When the `generate` command tries to generate the index creation script, it displays this message instead of generating a bad create index SQL statement.

ERROR: Table '*name*' which has links is not using `my_aa` as the `primary_key`

This error can occur during a `process` or `clone` command and indicates that the specified table contains link items but is not using `my_aa` as the key. You cannot set `active=0` or `item_key=0` for the `my_aa` column for a table that contains links. The AA Value is what allows the client to associate a LINK\_AI record with the corresponding record in the table, whose link needs to be updated.

ERROR: Table number *nnn* referenced in filter file does not exist

This message only occurs when there is an OCCURS table filter file "`dbfilter.cfg`" present in the config subdirectory for the data source. It indicates that the filter contains a reference to a table number that no longer exists. The most likely cause of this error is that the filter is not current. You should delete the binary filter file "`dbfilter.cfg`" from the config sub-directory and rerun the **makefilter** import command to recreate the file. If it this does not work contact Micro Focus Technical Support.

ERROR: Tables for DataSet *name*[/*rectype*] are not current; you must first run a `redefine` command

This message, which can occur at the start of a `process` or `clone` command, indicates that the layouts of the tables mapped from the data set are not current; therefore, they need to be updated via the `redefine` command. The DS\_Needs\_Redefining bit of the `status_bits` column of the DATASETS client control table keeps track of this.

ERROR: Tables for DataSet *name*[/*rectype*] need to be mapped; you must first run a `redefine` command

This message, which can occur at the start of a `process` or `clone` command, indicates that the mapping of the data set to relational database tables was not performed; therefore, you need to run a `redefine` command. The DS\_Needs\_Mapping bit of the `status_bits` column of the DATASETS client control table is used to keep track of this. This situation typically occurs if you try to run a `process` or `clone` command after setting the `active` column to 1 for a data set that was not previously mapped.

ERROR: The binary configuration file is not compatible with the client being used



The binary configuration file now has an additional parameter in the signon section that identifies the client type for which it was created. If you try to use this configuration with a different client, you will get this error (for example, if you try to use a Flat File Client configuration file with a SQL Server Client).

ERROR: The configured number of stmts (*mmm*) is insufficient for *nn* threads

Multi-thread updates need a greater number of configured database API statements (STMT) because of the increased number of concurrently executing SQL operations. The minimum allowed value is 20+ the number of threads.

ERROR: Time must be specified as '*hh:mm*', legal ranges are 0 to 23 for *hh* and 0 to 59 for *mm*

This message can occur in response to a command from the command-line console and indicates that the time specification values are invalid.

ERROR: Trace file prefix is too long, maximum allowable length is 20, value truncated

You can specify a prefix for the Client trace files in the Client configuration file, however this prefix is limited to 20 characters. If you use a longer prefix, you get this error. The default prefix is "trace".

ERROR: Tswitch command failed

The TSwitch (trace switch) command is a dbutility console command that closes the current trace file and opens a new file. If an IO error occurs during this operation this error is displayed. For more information, see "Log and Trace Files" in Appendix A of the *Databridge Client Administrator's Guide*.

ERROR: Unable to access registry key SOFTWARE\Micro Focus\Databridge\Client\6.6

The Client needs to access the Windows Registry key created by the installer in order to be able to run. If you did not use the installer and tried to copy the files, you will not get very far. Do not attempt to change the registry keys created by the installer as this might result in the Client being unable to operate.

ERROR: Unable to allocate *nnnn* bytes of memory

This message can occur during almost all Client commands. It indicates that the operating system does not have enough memory for various Client structures. The most common occurrence of this message is while loading the Client control tables. If this error occurs, do the following:

- ◆ Make sure that your system meets the minimum memory requirements for the hardware and software.
- ◆ Check the size of your swap file. The swap file could be too small or you could be running out of disk space on the volume where the swap file is located.
- ◆ Try again after quitting all other applications.
- ◆ Reboot the server if all else fails.

ERROR: Unable to convert DMSII type number to [a numeric] time for item '*name*' in table '*name*', {record will be discarded | time set to NULL} - Keys: *columnname* = *value*, ...

This message can occur during a `process` or `clone` command when you map a DMSII ALPHA or REAL item to a relational database integer data type (or in the case of SQL Server, a time data type). The `dms_subtype` specified must be one of the values defined in "Decoding DMSII Dates, Times, and Date/Times" in Chapter 2 of the *Databridge Client Administrator's Guide*.

```
ERROR: Unable to convert DMSII type number to a date for item 'name' in table
'name', {record will be discarded | date set to NULL} - Keys: columnname = value,
...
```

This message can occur during a `process` or `clone` command when you map a DMSII NUMBER or ALPHA item to a relational database date type. The `dms_subtype` (DMSII type *number*) specified must be one of the values defined in "Decoding DMSII Dates, Times, and Date/Times" in Chapter 2 of the *Databridge Client Administrator's Guide*.

```
ERROR: Unable to create "name" directory, errno=number (errortext)
```

This message, which can occur during a `dbutility configure` or `define` command or when customizing a new data source using the Client Configurator, indicates that the Client was unable to create the specified directory. (These directories include `config`, `logs`, `dbscripts`, `discards`, and `scripts` subdirectories.) You also get this message when the client tries to create the locks subdirectory in the service's working directory and the operation fails.

```
ERROR: Unable to create backup user script directory "path", errno=number
(errortext)
```

This message, which can occur during a `createscripts` command, indicates that the Client was unable to create the backup user script directory. In some cases this is simply a configuration error; check the configuration parameter `user_script_bu_dir` to see if it mistyped.

```
ERROR: Unable to create connection to database name for loading table 'tablename'
```

(SQL Server Client only) This error indicates that the Client was unable to create an ODBC connection for loading the given table using the BCP API. The only possible cause for this error is that the system does not have the resources. You should not encounter this message.

```
ERROR: Unable to create DACL, using default security
```

(Windows only) This message, which is only applicable when file security is enabled, indicates that the Client was unable to create an ACL to set up the security for a file or directory that it is trying to create. When this operation fails, the Client reverts to using default security. The Client sets up the Working Directory and its subdirectory with inheritance enabled so that if a file or sub-directory is created using default security, it inherits the security from its parent directory. This ensures that the files the user copies into the working directory (files created by the bulk loader) also have security enabled.

```
ERROR: Unable to create working directory "path", errno=number (errortext)
```

This message, which can occur during a `dbutility configure` or `define` command, indicates that the Client was unable to create global working directory for the client. The Client requires this is the location of the locks subdirectory.

```
ERROR: Unable to detect database name; specify it in the configuration file
```

This error message is limited to Oracle Express when using the default database. You normally should be able to use the client without specifying the database in the configuration file. You will get this message if the attempt to read the default database name fails.

```
ERROR: Unable to drop Databridge Client control tables
```

This message can occur during a `dbutility dropall` command. It indicates that an error occurred while dropping the client control tables. In this case, check the following:

- ◆ See the ODBC message that precedes this message (on the screen or in the log file) for more information. For Oracle, see [“OCI Errors” on page 157](#). For Microsoft SQL Server, see [“ODBC Errors” on page 156](#).
- ◆ Look at the previous output messages to see how far the Client progressed before encountering the error.

**NOTE:** After this message appears, you cannot rerun the `dbutility dropall` command if some of the Databridge control tables were dropped. You might have to use a relational database query tool to drop the remaining tables.

ERROR: Unable to expand block to *number* bytes

This message can occur when the client tried to expand a previously allocated memory block and the operation fails. It indicates that the operating system does not have enough memory for various Client structures. In this case, do the following:

- ◆ Quit all other applications and try again.
- ◆ Check the hardware and software requirements to make sure your system meets at least the minimum memory requirements.
- ◆ Check the size of your swap file. The swap file could be too small or you could be running out of disk space on the volume where the swap file is located.
- ◆ Reboot the server if all else fails.

ERROR: Unable to extract data for variable format date for item 'name' in table 'name', {record will be discarded | date set to NULL} - Keys: *column\_name* = *value*, ...

This message can occur during a `process` or `clone` command. It indicates that the client was unable to extract the various components of the variable format date format specified by the `dms_subtype`. The most likely cause of this error is that the number you entered is incorrect. Refer to "Unique DMSII Date/Time represented as ALPHA or NUMBER" in Chapter 2 of the *Databridge Client Administrator's Guide*.

ERROR: Unable to find base data set with `strnum = nnn` for virtual data set *name*[/*rectype*]

This message which can occur during a `define` or `redefine` command when you have virtual data sets and the parameter `automate_virtuals` is set to `True` and you are not using a MISER database. It indicates that the remote procedure call to the get the base structure index for the virtual data set failed. The most likely cause of this error is that the virtual data set is not properly defined.

ERROR: Unable to find control tables for DataSource *name* in file "*name*"

This error, which can occur during a `reload` command, indicates that the archive file does not contain any entries for the data source on the command line. You either mistyped the data source name on the command line, you are not using the correct archive file, or you did not back up the data source you are trying to reload.

ERROR: Unable to find DataSource *name*

This message can occur during all Client commands except `configure`, `define`, and `dropall`. It indicates that you entered a data source name that is not in the DATASOURCES client control table. This can occur if the data source name is misspelled or you have not created the data source yet.

ERROR: Unable to find matching concat data item (*nnn*) record for item '*name*' in table '*name*'

This message indicates that the `dms_concat_num` column in `DMS_ITEMS` contains an incorrect value. This value may refer to a nonexistent or inactive column. The user script involved is most likely the cause of the error and should be examined. Avoid using hard-coded number in user scripts; instead, use sub-queries.

ERROR: Unable to find matching data item record for DMS Item Number *nnn* in table '*name*'

This error can occur during any Client command that loads the client control tables when the data source contains an active data set that has active item with an OCCURS DEPENDING ON clause. The loading of the client control tables dynamically sets up the links between the given item and the item on which the OCCURS clause depends. This link may go back to the primary table if the OCCURS item is in a secondary table. The `dms_item_number` column is used as a foreign key. If the load cannot find such an item in the `DATAITEMS` table this message is displayed.

The only possible cause for this error is the active column of the `DMS_ITEMS` table of the depends item set to 0, causing it vanish from the `DATAITEMS` table. To resolve this error, set the active column to 1, set the `DS_Needs_Remapping` bit in the `ds_options` column of the corresponding `DATASETS` client control table, and execute a `redefine` and a `generate` command.

ERROR: Unable to get stmt for {insert | delete | update | delete all} statement for table '*name*'

This message, which can occur during a `process` or a `clone` command, indicates that the Client was unable to get a statement (OCI, ODBC or CLI) for executing the specified SQL statement. This message is typically preceded by a database API error message. This message will only occur when the configuration parameter `use_stored_procs` is set to `False`. In this case the client generates `INSERT`, `UPDATE` and `DELETE` statements instead of calling stored procedures to execute these statements. Both methods use host variables, however not using stored procedures is more efficient, but it uses a bit more memory to hold the SQL statement, which is quite a bit longer. The "delete\_all" case applies to OCCURS tables where all the rows for a given key are deleted in one SQL statement by calling the stored procedure `z_tablename`.

ERROR: Unable to get stmt for stored procedure '{i|u|d|z}\_name'

This message, which can occur during a `process` or a `clone` command, indicates that the Client was unable to get a statement (OCI, ODBC or CLI) for executing the specified stored procedure. This message is typically preceded by a database API error message.

ERROR: Unable to get stmt for updating control table *name*

This message, which can occur during most commands, indicates that the Client was unable to get a statement (OCI, ODBC or CLI) for executing an update statement for the corresponding Client control table. This message is typically preceded by a database API error message.

ERROR: Unable to handle sql\_type *dd* for external column '*name*' in table '*name*'

This message occurs during a `process` or `clone` command and indicates that the Client cannot handle the `sql_type` for the external column being added to the specified data set.

ERROR: Unable to load backup copy of control tables for DataSource *name*

During Client Configuration operations, the initial state of the control tables is automatically saved in the unload file *src\_reorg\_ddd.cct* where *src* is the data source name and *ddd* is the update level of the database. Client Configuration operations run much like *redefine* commands. When the Client Configurator needs to compare the old and the new layout it reloads the old control tables from this file.

**CAUTION:** Deleting the unload file is not recommended until you have completed all customization tasks in the Client Configurator. The Client automatically deletes the unload file when it no longer needs it. If you run the Client Configurator multiple times before running a *process* command, subsequent executions will not back up the control tables. The Client Configurator needs the unload file (that is, the original backup) to determine which changes have been made.

When you run back to back *redefine* commands that do not require a *reorg* command the client will reload the control tables from the backup file it creates instead of using the updated tables, in case you intend to rerun the command (e.g. after fixing a user script that was incorrect). If this operation fails, you will also get this error. Look at the log file and the unload file to see if you can find out why the operation failed.

ERROR: Unable to load data source list

This message occurs during a *define* or *redefine* command indicates that the Client cannot load the data source list. When you have more than one data source in the same relational database, the Client uses the data source list to find out the table names used by other data sources and prevent naming conflicts.

ERROR: Unable to locate DLL "*name*"

(SQL Server Client only) When using the BCP API, the Databridge Client loads the ODBC DLL and sets up a table of address through which it makes the BCP API calls. This allows the Client to work with the version of the ODBC DLL that supports the features needed. This error indicates the client cannot find the specified DLL. Make sure that the directory where the DLL resides is in the system PATH.

ERROR: Unable to locate the extended translation {DLL | shared library} "*filename*"

The attempt to locate the external data translation DLL, DBEATRAN.DLL, using a *LoadLibrary* call failed. Windows looks for a DLL in several places, the first of which is the directory where the program being executed resides. Under normal circumstances, this is the program directory created by the install process of the Client (c:\Program Files\Micro Focus\Databridge\6.6\*dbase\_type*). If you use the Japanese version of the DLL, select the appropriate feature, in the **Feature Selection** tab of the installer to copy the DLL and sample configuration files to this directory. If you use a different DLL, we recommend that you move the DLL to this directory. Windows also looks for the DLL in the current directory, the Windows system directory, the Windows directory, and the directories listed in the PATH environment variable. If the DBEATRAN DLL is in none of these places, this error message is displayed when you set the configuration parameter *use\_ext\_translation* to Yes.

On UNIX, the environment variable *LD\_LIBRARY\_PATH* must contain the directory where the shared library resides. On some 32-bit systems (for example, *SHLIB\_PATH* for HP-UX and *LIBPATH* for AIX), a different environment variable is used.

ERROR: Unable to read first source record from archive file "*name*", errno= *number* (*errortext*)

This error, which can occur during a `reload` command, indicates that after successfully reading the version record, the client got an I/O error when it tried to read the next record, which should be a data source record. The most likely cause of this error is a corrupt unload file.

ERROR: Unable to read translation configuration files and initialize tables

This message, which only occurs at the start of `process` or `clone` command when using an external translation DLL (or shared library), indicates that the translation DLL initialization was unsuccessful. In most cases, the DLL can not find the translation configuration file. The DLL expects these configuration files to be located in the config directory where the client configuration files reside.

ERROR: Unable to read version record from archive file "*name*", *errno=number* (*errortext*)

This message can appear if a file I/O error occurs during a `reload` command when trying to read the first record of the file (which is always *V,version*). The included system error should give you more information about why this error occurred.

ERROR: Unable to retrieve database/server names from data source

The SQL Server Client uses `SQLGetInfo` calls to programmatically retrieve the database and server name for the ODBC data source, which eliminates the need to specify the names in the configuration files. This message indicates that there was an error while retrieving these names. Check the preceding ODBC error message for more information about the reason for the failure.

ERROR: Unable to retrieve value for 'FileSecurity' from registry (result = *nnn*)

This error is an indication that the Windows registry keys for the Client are corrupt. The Client expects to find the string `FileSecurity` in the Windows Registry key `HKEY_LOCAL_MACHINE\SOFTWARE\Micro Focus\Databridge\Client\6.6` where the installer saves several values including the name of the directory in which the software was installed. The name `FileSecurity` is a `REG_DWORD`, whose value is 1 or 0. This value determines whether file security is enabled or not. This error indicates that there is no such entry in the list of values for above mentioned registry key. To resolve this issue, change the setting for `FileSecurity` using the `setfilesecurity` program included with `Databridge`; do not edit the registry key with `regedit`.

ERROR: Unable to retrieve ODBC driver name/version from data source

(SQL Server Client only) When starting up the first ODBC connection SQL Server Client makes a couple of ODBC calls to get the ODBC driver name and version. This information is critical in being able to use the BCP API. If you get this error, make sure that you are using one of the recommended ODBC drivers, whose name is of the form `MSODBCDLL1x.dll`. If you are using one of these drivers and the problem persists try rebooting the server and if that fails contact Micro Focus Customer Care.

ERROR: Unable to retrieve value for 'INSTALLDIR' from registry (result = *nnn*)

This error indicates that the Windows Registry keys for the Client are corrupt. The Client expects to find the string `INSTALLDIR` (the name of the directory in which the software was installed) in the Windows Registry key `HKEY_LOCAL_MACHINE\SOFTWARE\Micro Focus\Databridge Client\6.6`. The installer saves several values to this key, including the name of the directory in which the software was installed. This error indicates that there is no such entry in the list of values for above mentioned Registry key.

ERROR: Unable to retrieve value for 'UserID' from registry (result = *nnn*)

This error is an indication that the Windows Registry keys for the Client are corrupt. The Client expects to find the string `UserID` in the Windows Registry key `HKEY_LOCAL_MACHINE\SOFTWARE\Micro Focus\Databridge Client\6.6`. The installer saves the name of the directory where the software was installed and several other values in this Registry key.

ERROR: Unable to retrieve value for 'WORKINGDIR' from registry (result = *nnn*)

This error indicates that the Windows Registry keys for the Client are corrupt. The Client expects to find the string `WORKINGDIR` (the working directory for the service and location of the locks subdirectory) in `HKEY_LOCAL_MACHINE\SOFTWARE\Micro Focus\Databridge Client\6.6`. For more information about the working directory, see the *Databridge Installation Guide*.

ERROR: Unable to update specified external column *nn*

This error can only occur when processing a binary configuration file that contains `external_column` specifications and indicates that the Client was unable to update its internal table for external column definition. If using a text configuration file, check the syntax of the `external_column[nn]` line. If using a binary file, export the configuration file, verify the syntax, (make any corrections), and then import the file. The `import` command also checks the syntax for you.

ERROR: Undefined date format type (*number*) for item 'name' in table 'name',  
{record will be discarded | date set to NULL} - Keys: *columnname = value, ...*

This message indicates an invalid value in the `dms_subtype` column in the `DATAITEMS` Client control table. For a list of date formats refer to Decoding DMSII Dates, Times, and Date/Times in Chapter 2 of the *Databridge Client Administrator's Guide*.

ERROR: Undefined section header in configuration file line: *text*

See "Sample SQL Server Client Configuration File" in Appendix C of the *Databridge Client Administrator's Guide* for valid section headers.

ERROR: Unimplemented command *cmd\_number*

This internal `DBCIntCfgServer` error indicates that the Console or Client Configurator attempted to execute an unimplemented RPC. This can only happen if you try to run a newer version of the Console or Client Configurator with an old Client.

ERROR: Unknown command: *command*

This message can appear when you misspell a `dbutility` command, or you enter a command that does not apply to `dbutility`.

ERROR: Unknown console command; type "help" to get a list of commands

This message indicates that the operator entered an invalid command in the command-line console.

ERROR: Update of *colname* column in `DATASETS` table failed for *name[/rectype]*

This message, which can occur during a `process`, `clone`, `define` or `redefine` command, indicates that an attempt to update the specified column of the DATASETS Client control table failed. The columns in question include the following columns:

- ◆ `active`
- ◆ `ds_mode`
- ◆ `misc_flags`
- ◆ `status_bits`
- ◆ `ds_options`

Check the preceding SQL error message to determine why the error occurred.

```
ERROR: Update of DATAITEMS entries failed for table 'name' [for DataSet name[/rectype]]
```

This error can occur during a `redefine` command or when running the Client Configurator. It indicates that the update of the DATAITEMS control table failed when trying to restore customizations. See the SQL errors that precede this message to find out why this error occurred.

```
ERROR: Update of DATAITEMS table failed for 'name' of table 'name'
```

This error can occur when running the Client Configurator or when running a `redefine` command. If it occurs when running the Client Configurator, it indicates that DBCIntCfgServer was unable to update the DATAITEMS control table for the specified item of the given table, while applying customization changes made by the user. The `redefine` command can get this error whenever it tries to update the DATAITEMS table and something goes wrong. See the preceding SQL errors to determine why this error occurred.

```
ERROR: Update of DATASETS table failed for Global_DataSet
```

This message can occur during a `redefine`, `clone`, or `process` command. It indicates that an error occurred when updating the DATASETS Client control table for the `Global_DataSet`. See preceding API error messages (onscreen or in the log file) to determine the reason for this error.

```
ERROR: Update of DATASETS table failed for name[/rectype]
```

This message can occur during a `process` or `clone` command when the DATASETS Client control table is updated for a given data set (the `/rectype` is added for variable-format data sets except type 0 records). This error can occur during the cloning of a data set when the `stateinfo` is changed multiple times and at the end of a `process` command when the global `stateinfo` is being propagated to all data sets whose `in_sync` column has a value of 1. This message can occur when running the Client Configurator. See the database API messages that precede this error for more information.

```
ERROR: Update of DATASOURCES table failed for name
```

This message can occur whenever the DATASOURCES client control table is updated for a given data source. This can happen at the beginning and end of a `process` or `clone` command when the `status_bits` column is updated. See the database API messages that precede this error for more information.

```
ERROR: Update of DATATABLES and DATAITEMS tables to preserve pass2 changes failed for DataSet name[/rectype]
```



This message, which can occur during a `redefine` command or when running the Client Configurator, indicates that the Client could not update the DATATABLES and DATAITEMS entries while attempting to preserve changes. Check the SQL error message that precedes this message as to why this error occurred.

ERROR: Update of DATATABLES failed for table '*name*' [for DataSet *name*[/*rectype*]]

This message, which can occur during most commands, indicates that the Client could not update the DATATABLES entry for the specified table. Check the SQL error message that precedes this message as to why this error occurred.

ERROR: Update of DMS\_ITEMS table failed for *name* for DataSet *name*[/*rectype*]

This message, which can occur during a `redefine` command or when running the Client Configurator, indicates that the Client could not update the DMS\_ITEMS entries. Check the SQL error message that precedes this message for clues about why this error occurred.

ERROR: Update of DMS\_ITEMS table failed for *name* (item\_number *nnn*) in DataSet *name*[/*rectype*]

This message indicates that the DBCIntCfgServer was unable to update the DMS\_ITEMS Client control table when processing an update request from the Client Configurator. Check the SQL error message that precedes this message as to why this error occurred.

ERROR: Update of DMS\_ITEMS table to preserve *pass1* changes failed for DataSet *name*[/*rectype*]

This message, which can occur during a `redefine` command or when running the Client Configurator, indicates that the Client could not update the DMS\_ITEMS entries while preserving previous customizations. Check the SQL error message that precedes this message as to why this error occurred.

ERROR: Update statistics failed for table '*name*'

This message applies to the Client for Microsoft SQL Server. It can occur during a `process` or `clone` command after an index for a table is created. The `update statistics` SQL Server command causes the software to update the statistics on the current table. This message indicates that the `update statistics` command has failed.

ERROR: Update Worker thread[*nn*]: `bcp_done()` failed for table '*name*'

(SQL Server Client only) The SQL Server Client issues a `bcp_done` call into the BCP API when it receives a State Info record from the Databridge Engine that has a mode of 1 at the end of data extraction. This call commits the current batch of rows and instructs the BCP API that we have reached the end of the data for the load. This error usually means that the database is out of resources. Check any ODBC errors that precede this error for clues about what happened.

ERROR: Update Worker thread[*nn*]: `BCP_Record()` failed

(SQL Server Client only) This error can occur when you use multi-threaded updates, and indicates that the attempt to load a row using the BCP API failed. Reference the ODBC errors in the log file to see if you can get any clues as to why this error occurred. If the error persists try setting the parameter `n_update_threads` to 0 or use the `-l` option to force the client to use the bulk loader (`bcp`) or use the `-l` option to force the client to use the bulk loader (`bcp`).

ERROR: Update Worker thread[*nn*]: `FinishDataExtraction` failed for table '*name*'

(Oracle Client or SQL Server Client using bcp only) When processing multi-threaded extracts the Client queues a working storage block which marks the end of extraction. The thread calls the procedure **FinishDataExtraction**.

In the case of Windows, where the client uses temporary files, this involves closing the temporary file and queuing it on the bulk-loader thread's work queue. This error usually indicates that the close of the temporary file has failed. In the case of UNIX clients, which use a pipe to run the bulk loader, this error usually indicates a problem with the actual load. Refer to the sql\*loader log in the data source working directory for details on why this error occurred.

ERROR: Update Worker thread[nn]: Illegal response type rr encountered

This is an internal error that can only occur when using multi-threaded updates it indicates that an Update Worker encountered a DMS buffer whose record type field is illegal. The only record types that are expected by the Update Worker are CREATE, DELETE, MODIFY, MODIFY\_AI, LINK\_AI, or DELETE\_ALL any other record types are handled in the main thread. Report this error to Micro Focus Technical Support.

ERROR: Update Worker thread[nn]: {Read\_CB\_CREATE() | Read\_CB\_DELETE() | Read\_CB\_DELETE\_ALL() | Read\_CB\_MODIFY() | Read\_CB\_LINKS()} failed

This error can occur when you use multi-threaded updates, it indicates that the operation in question failed. Look at the SQL errors in the log file to see if you can get any clues as to why this error occurred. If the error persists try setting the parameter `n_update_threads` to 0.

ERROR: User columns of dms\_subtype n1 and n2 are mutually exclusive

This message can occur when processing text configuration files. It indicates that two of the user columns you specified are mutually exclusive. For example 1 (`update_type`) and 11 (`expanded update_type`) or 10 (`deleted_record`) and 11 (`expanded update_type`).

ERROR: User columns of dms\_subtype mmm can only be used in conjunction with dms\_subtype mmm

This message can occur when processing text configuration files. It indicates that you are attempting to use the external column `delete_seqno` when the column `deleted_record` is not present. The column `delete_seqno`, which allows more than one delete to be performed for the same record when the second clock remains unchanged, is only meaningful when the column `deleted_record` is present. In the absence of the `delete_seqno` column the Client cannot make the update and stalls until the clock changed, thereby degrading performance.

ERROR: User DataSet {define | layout} script "filename" failed

This message can appear during a `define` or `redefine` command. It indicates that the specified user script failed. Check API or SQL errors that precede this message to see why the script failed. If you cannot figure out why it is failing, use the `-t 0x800` option for the command or use the `runscript` command to test the script.

ERROR: User DataSets global layout script "filename" failed

This message can appear during a `define` or `redefine` command. It indicates that specified user script failed. Check API or SQL errors that precede this message to see why the script failed.

ERROR: User DataTables global define script "filename" failed

This message can appear during a `define` or a `redefine` command and indicates that the specified user script failed. Check API or SQL errors that precede this message to see why the script failed. This message may appear after other related error messages that need to be fixed first.

ERROR: User global layout script "filename" failed

A new type of user script that applies to all data sources was added to the 6.5 Client for Miser customers that shares the same scripts with different data source names. This script, which is named "**script.user\_all\_sources**", is run before the global data sets scripts are run. This error indicates that the specified script failed. Check API or SQL errors that precede this message to see why the script failed.

ERROR: User script "filename" failed

This message can appear during a `runscript` command and indicates that the specified script failed. Check API or SQL errors that precede this message to see why the script failed.

ERROR: User stored procedure creation script "name" failed

This message indicates that the client got an error running the user script `script.user_create_sp.name`, which is used to split up the actions that would normally be executed in the user script `script.user_create.name`. This last user script is executed at the time a table is created at the beginning of the data extraction phase. The stored procedure user script is also executed when a `refresh` command is executed. This command gets automatically run during the execution of a `reorg` command. This allows user written stored procedures to be kept current after a DMSII reorganization, rather than just creating them and finding out that they no longer work after a DMSII reorganization.

ERROR: User table creation script "filename" failed

This message can appear during a `process` or `clone` command and indicates that specified data table creation user script fails. Check API or SQL errors that precede this message to see why the script failed.

ERROR: Value out of range for 'F' option argument

The F option is used to pass dbutility an AFN after which it stops. If you use a value that is not in the range 1 to 9999 this error is displayed.

ERROR: Value out of range for 'V' option argument - using *nn* instead

The V option is used to pass dbutility a control table version to use in the `unload` command. The value being passed must be greater than 0 and less than or equal to the present version of the client control tables. For 6.6 clients the client control table version is 31; for the 6.5 client the client control table version is 29, and for 6.2 and 6.3 clients, the client control table version is 26.

ERROR: Virtual data set link for data set *name[/rectype]* is NULL, make sure that `automate_virtuals` is true

In a MISER database, user scripts link the virtual data sets and the real data sets from which they are derived by using the `virtual_ds_num`, `real_ds_num` and `real_ds_rectype` columns in the DATASETS Client control table. If the configuration parameter `automate_virtuals` is not enabled, this pointer is not set up and executing a `createscripts` command returns this message.

For non MISER databases, the Client gets the information from the Engine when the parameter `automate_virtuals` is set to True. This makes the Client handle virtual data sets in a much more coherent manner by paying attention to the relation between virtual data sets and the actual data sets they are derived from.

ERROR: Work\_desc pool empty

This error, which only occurs with multi-threaded updates, indicates that there is a bug in the Client. The work descriptors are tiny records that are used to queue the same DMS buffer on multiple update worker work queues. Running out of these structures is an indication that Client is failing to return some of these to the pool when they are no longer needed.

ERROR: Write failed for binary configuration file "*name*", *errno=number (errortext)*

This message can occur during a `dbutility import` command or when `DBCIntCfgServer` updates a client configuration file. It indicates that an I/O error occurred while writing to the specified binary configuration file. The system error included in this message should explain why this error occurred.

ERROR: Write failed for bulk load file for table '*name*', *errno=number (errortext)*,  
Record: *recordtext*

This message applies only to Windows Clients. This message can occur during a `process` or `clone` command and indicates that an I/O error occurred while writing to the specified temporary data file. This message typically occurs when you run out of disk space (resulting in the *errortext* "Out of Disk Space"). The system error included in this message should explain why this error occurred.

ERROR: Write failed for discard file for table '*name*', *errno=number (errortext)*

This message, which can occur during a `process` or a `clone` command, indicates that an error occurred while writing a record to a discard file, whose name appears in the message. The system error included in this message should explain why this error occurred. The most common cause of this error is running out of disk space. Make sure you take advantage of the recently added configuration parameter `max_discards` that allows you to prevent this situation from occurring. You can do this in one of two ways: (1) you can either make the client abort after a certain number of discards, regardless of which table they belong to, or (2) you can limit the number of discards records written to a discard file for any given table.

ERROR: Write failed for log descriptor file "*name*", *errno=number (errortext)*

The Client uses the binary file `log.cfg` to keep track of the log filename. If the error persists, delete this file and the Client will create a new one. The most likely source of this error is file ownership conflicts between the command-line client, `dbutility`, and the service. The system error message included in this message should provide information about why this error occurred

ERROR: Write failed for Null Record file, *errno=number (errortext)*

This message, which can occur during a `define` or `redefine` command, indicates that an I/O error occurred while writing to the null record file. The system error included in this message should explain why this error occurred.

ERROR: Write failed for pipe for table '*name*', *errno=number (errortext)*, Record:  
*recordtext*

This message, which is limited to UNIX, can occur during a `process` or a `clone` command when the main process is writing data to a UNIX pipe used to pass data records to the bulk loader process that runs the Oracle SQL\*Loader utility *errortext* provides information as to why the error occurred. The most common cause of this error is the bulk loader exceeding the maximum discards threshold and aborting. In this case, the *errortext* will be "Broken Pipe".

ERROR: Write failed for trace descriptor file "*name*", *errno=number (errortext)*

The Client uses the binary file `trace.cfg` to track the trace filename. If this error persists, delete this file and the Client will create a new one. The most likely source of this error is file ownership conflicts between the command-line client, `dbutility`, and the service. The system error message included in this message should provide information about why this error occurred.

ERROR: You need to run a `reorg` command for DataSource *name*

This message indicates that you are trying to run a `redefine` command after a successful run of the command that determined that a `reorg` command need to be run next. If you do not want to run the `reorg` command, you must reload the control tables form the backup file created by the `redefine` command.

ERROR: 'Y' option must be followed by the text 'reclone\_all'

When you use the `-Y` option with the command line Client `dbutility`, you must specify the argument 'reclone\_all' after `-Y`. Failure to do so results in this error, which is meant to prevent accidental use of `-Y`, when you meant to type `-y`.

RPC PROTOCOL ERROR: Type = *responsetype* StrNum=*number*

This message can occur during any Client command that involves communications with DBServer or Enterprise Server. It indicates that an RPC protocol error occurred while trying to read a response packet to an RPC. In this case, try again. If this error occurs persistently and is reproducible, contact Micro Focus Technical Support.

## Usage Errors

Usage error messages appear onscreen and in the log file in response to incorrect syntax or missing parameters when you enter `dbutility` commands. If tracing is enabled, messages are also captured in the trace file. These messages are prefixed by the timestamp `hh:mm:ss` (where *hh* is hours, *mm* is minutes, and *ss* is seconds)

Each usage message indicates the correct syntax and parameters, as follows:

```
USAGE ERROR: dbutility [options] clone <DataSource> <DataSets>
USAGE ERROR: dbutility [options] configure
USAGE ERROR: dbutility [options] createscripts <DataSource>
USAGE ERROR: dbutility [options] define <DataSource> <host> <port>
USAGE ERROR: dbutility [options] display <DataSource>
USAGE ERROR: dbutility [options] drop <DataSource>
USAGE ERROR: dbutility [options] dropall
USAGE ERROR: dbutility [options] export [<File>]
USAGE ERROR: dbutility [options] generate <DataSource>
USAGE ERROR: dbutility [options] import [<File>]
USAGE ERROR: dbutility [options] process <DataSource>
USAGE ERROR: dbutility [options] redefine <DataSource>
USAGE ERROR: dbutility [options] refresh <DataSource> <DataSet>
USAGE ERROR: dbutility [options] reload <DataSource> <File> [<DataSet>...]
USAGE ERROR: dbutility [options] reorg <DataSource>
USAGE ERROR: dbutility [options] rowcounts <DataSource>
USAGE ERROR: dbutility [options] runscript <ScriptFile>
USAGE ERROR: dbutility [options] switchaudit <DataSource>
USAGE ERROR: dbutility [options] tcptest <DataSource> [host port]Len Count
USAGE ERROR: dbutility [options] unload <DataSource> <File>
```

For information on dbutility commands and options, see Appendix B of the *Databridge Client Administrator's Guide*.

## Host Messages That Appear on the Client

The following host messages may appear onscreen and in the log file of the Client. If tracing is enabled, messages are also captured in the trace file.

All host messages are followed by a second message that provides an error number.

```
HOST ERROR: hosterrortext  
Error Text: text
```

where *hosterrortext* is a message from the Client, as listed in this section.

Where	Is
<i>hosterrortext</i>	A message from the Client, as listed in this section. Host messages are prefixed by the timestamp <i>hh:mm:ss</i> , where <i>hh</i> is hours, <i>mm</i> is minutes, and <i>ss</i> is seconds. This message typically identifies the API (Engine entry point) being called and provides additional about the task being performed, whenever this is relevant.
<i>text</i>	A message from the DBEngine or DBServer on the host, or Enterprise Server. All such messages contain a number whose range determines the origin of the message. DBEngine messages start at 0000, DBServer messages start at 1000, and Enterprise Server messages start at 1100.

```
HOST ERROR: Databridge call failed for ATMEcho, error_number = number
```

The Client uses this API to verify that the transport layer is operational. This is always the first RPC; DBServer echoes the test pattern back to the Client. This message indicates that the Client received an unrecoverable error. The Client recovers from a timeout error or from receiving bad data by disconnecting and reestablishing the connection. When the `-v` option is used, the Client displays the message "Disconnecting and restarting transport initialization" for timeouts.

```
HOST ERROR: Databridge call failed for baselist, error number = number
```

The Client uses this API to get the structure numbers of the real data sets from which a virtual data set is derived. It only does this when the parameter `automate_virtuals` is set to True and the parameter `miser_database` is set to False.

```
HOST ERROR: Databridge call failed for connect, error number = number
```

The Client is unable to connect to the Databridge Server (DBServer or DBEnterprise Server). Make sure that the host name is correct (that is, it matches the host name in the DATASOURCES table) and the domain name server is running.

—or—

Change the host name in the DATASOURCES table to the IP address and try the Client command again. Verify that the port number matches the port number in the DBServer parameter file and that the mainframe, the Databridge Engine, and DBServer are running. If you are connecting to DBEnterprise Server, you also need to make sure that the DBDirector service is running.

HOST ERROR: Databridge call failed for datasets, error number = *number*

The Databridge Engine was unable to retrieve the requested information on data sets. This message can occur when you run a `define` or `redefine` command and get an error from the Databridge Engine or the Databridge Server.

HOST ERROR: Databridge call failed for DB\_ClientKey on DataSet *name[/rectype]*, error number = *number*

The Client was unable to register the Client keys with the Databridge Engine for the specified data set, which has the KEYCHANGEOK attribute.

HOST ERROR: Databridge call failed for DB\_DeSelect on DataSet *name[/rectype]*, error number = *number*

This error can occur during a `process` or `clone` command. It indicates that the Client attempted to request that the Databridge Engine stop sending updates for the data set (or data set record type combination in the case of a variable-format data set) and that the RPC failed.

HOST ERROR: Databridge call failed for DB\_PumpMsg, error number = *number*

DB\_PumpMsg is an RPC to DBServer on the host that creates data traffic to test the speed of the TCP/IP connection. This error can occur during a `tcptest` command and indicates that the RPC failed.

HOST ERROR: Databridge call failed for DB\_Read call, error number = *number*

This message indicates that an error occurred during a `process` or `clone` command for the specified DB\_Read RPC, which is used to request updates from the Engine.

HOST ERROR: Databridge call failed for DB\_Select on table *datasetname*, error number = *number*

This error can occur during a `process` or `clone` command, and it indicates that the Databridge Engine was unable to select the requested data set. As part of initialization, the Client send a DB\_Select request for every data that it to be cloned or tracked. The DB\_Select RPC data include the stateinfo for the data set, which includes the mode and the audit location from which to resume tracking changes. If the data set is to be cloned, the stateinfo consists of all zeroes.

HOST ERROR: Databridge call failed for DB\_Wait call, error number = *number*

This message indicates that an error occurred during a `process` command for the specified `DB_Wait` RPC. This message appears only if the `use_dbwait` parameter is set to true in the client configuration file (or you use the `/w` command line switch to toggle it to true). Instead of returning a `DBM_AUD_UNAVAIL (7)` status to the Client, the Engine enters a wait-and-retry loop. It retries every `max_retry_secs` until either more updates become available, or, `max_wait_secs` elapses (without any updates becoming available). In the case of the latter, the Engine sends the Client a `DBM_AUD_UNAVAIL (7)` status, which causes the Client run to terminate.

HOST ERROR: Databridge call failed for `dbinfo`, error number = *number*

This message, which can occur when you run a `define` or `redefine` command, indicates that the Client was unable to retrieve information about the DMSII database from the Databridge Engine using the `DB_Info` RPC. This error can also occur during a `process` or `clone` command. The Client uses this RPC to determine if the update level of the DMSII database has changed and makes the appropriate change to the update level value in the `DATASOURCES` Client control table when the Client ignores an update level change. It also uses this information to inhibit clones during periods where there is a DMSII reorganization that the client has not yet caught up to.

HOST ERROR: Databridge call failed for `initialize`, error number = *number*

The Databridge `initialize` remote procedure call could not initialize the session between `DBServer` and the Client. This message occurs after three unsuccessful attempts for a TCP/IP connection. This message can occur when you run a `define`, `redefine`, `process`, `clone`, or `switchaudit` command.

HOST ERROR: Databridge call failed for `initialvalue`, error number = *number*

The Client uses the `initialvalue` remote procedure call (RPC) to get the initial value for a data set. It uses this information to assign values to columns that are added as a result of a DMSII reorganization. This message only occurs during a `redefine` command.

HOST ERROR: Databridge call failed for `keys`, error number = *number*

Databridge was unable to retrieve information on keys. This message can occur when you run a `define` or `redefine` command.

HOST ERROR: Databridge call failed for `layout`, error number = *number*

Databridge was unable to retrieve information on the layout of a data set. This message can occur when you run a `define` or `redefine` command.

HOST ERROR: Databridge call failed for `nullrecord`, error number = *number*

The Client was unable to retrieve the `nullrecord` information for a data set. This message can occur during a `define` or `redefine` command when the parameter `read_null_records` is set to True. A `nullrecord` is a DMSII record for a data set where all items are set to NULL using the null values defined in the `DASDL`. The Client uses this information to construct the `nullrecord` file, which is used to determine if items are NULL.

HOST ERROR: Databridge call failed for `open_audit`, error\_number = *number*



The Client uses this API during the `redefine` command to force the Databridge Engine to use the correct description file by opening the audit file that was being read when the DMSII reorganization was detected. This technique makes it possible to process multiple reorganizations when the Client is far behind DMSII.

The Client also uses this API during `process` commands to ensure that the Engine uses the correct description file when processing DMSII data. When the client has not yet caught up with a DMSII reorganization, the proper description file must be used.

HOST ERROR: Databridge call failed for parameters `CONCURR_EXTR`, error number = *number*

The Client could not override the Databridge Engine parameter `WORKERS = n` during the execution of a `process` or `clone` command.

HOST ERROR: Databridge call failed for parameters `TG_BLOCKS`, error number = *number*

The Client could not override the Databridge Engine parameter `CHECKPOINT every nnn AUDIT BLOCKS` during the execution of a `process` or `clone` command.

HOST ERROR: Databridge call failed for parameters `TG_ELAPSED`, error number = *number*

The Client could not override the DBEngine parameter `CHECKPOINT every nnn SECONDS` during the execution of a `process` or `clone` command.

HOST ERROR: Databridge call failed for parameters `TG_TRANS`, error number = *number*

The Client could not override the Databridge Engine parameter `CHECKPOINT every nnn TRANSACTIONS` during the execution of a `process` or `clone` command.

HOST ERROR: Databridge call failed for parameters `TG_UPDATES`, error number = *number*

The Client could not override the Databridge Engine parameter `CHECKPOINT every nnn UPDATE RECORDS` during the execution of a `process` or `clone` command.

HOST ERROR: Databridge call failed for `primary_key`, error number = *number*

The Client was unable to retrieve information on the primary keys for a data set. This message can occur when you run a `define` or `redefine` command.

HOST ERROR: Databridge call failed for `primary_set`, error number = *number*

The Client was unable to retrieve information on the `primary_set` for a data set. This message can occur when you run a `define` or `redefine` command.

HOST ERROR: Databridge call failed for `reset_option LONGTRANS`, error number = *number*

The Client could not reset the Databridge Engine parameter `CHECKPOINT LONG TRANSACTIONS` during the execution of a `process` or `clone` command.

HOST ERROR: Databridge call failed for `server_option INHIBIT_WFL`, error number = *number*

The Client was unable to set the server option that inhibits the running of a WFL when an audit file is processed by the Engine. Normally such a WFL would be used to remove the audit files, once the Engine is done with them.

HOST ERROR: Databridge call failed for `server_option STOP_{BEFORE | AFTER}_NAME`, error number = *number*

The Client was unable to set the server option that causes the Engine to stop processing audit files when the specified task name is encountered in the audit trail. The Engine can stop at the quiet point before or after the specified task is encountered. These values are defined in the client configuration file using the parameters `stop before task` or `stop after task`.

HOST ERROR: Databridge call failed for server\_option STOP\_{BEFORE | AFTER}\_TIME, error number = *number*

The Client was unable to set the server option that causes the Engine to stop processing audit files when the specified time is reached. These values are defined in the client configuration file using the parameters `stop before task` or `stop after task`.

HOST ERROR: Databridge call failed for server\_stats, error number = *number*

The Client was unable to obtain the server statistics using the DB\_WorkerInfo RPC.

HOST ERROR: Databridge call failed for set\_option DOC, error number = *number*

Databridge could not set the DBServer option that requests DOC records to be sent during audit file processing. This message can occur at the beginning of a `process` or `clone` command when the Client uses the DBSetOption remote procedure call to request that the Databridge Engine provide documentation (DOC) records while reading the audit file.

HOST ERROR: Databridge call failed for set\_option DOC\_AFN, error number = *number*

Databridge could not set the DBServer option that requests DOC records to be sent when a new audit file is open during audit file processing. This message can occur at the start of a `process` or `clone` command. The Client uses the DBSetOption remote procedure call to request that the Databridge Engine provide documentation (DOC) records for all audit files when the Engine starts processing them.

HOST ERROR: Databridge call failed for set\_option Embedded Extract w/o IndependentTrans, error number = *number*

The Client could not set the Databridge Engine option that indicates that embedded data sets should be extracted even though INDEPENDENTTRANS is reset. Such data sets cannot be tracked.

HOST ERROR: Databridge call failed for set\_option GLOBAL\_SI, error number = *number*

The Client could not set the DBServer option that requests global stateinfo records to be sent during audit file processing instead of sending stateinfo records for each data set. The Engine only honors this request when all data sets are in mode 2 and they have identical stateinfo values (that is, they are all in sync).

HOST ERROR: Databridge call failed for set\_option ITEMCOUNT, error number = *number*

The Client could not set the Databridge Engine option that indicates it should use the new method for calculating the item counts of data sets.

HOST ERROR: Databridge call failed for set\_option LONGTRANS, error number = *number*

The Client could not set the Databridge Engine parameter CHECKPOINT LONG TRANSACTIONS during the execution of a `process` or `clone` command.

HOST ERROR: Databridge call failed for set\_option MAXRECS, error number = *number*

The Client was unable to set the DBServer MAXRECS option using the DB\_ServerOption RPC. The MAXRECS option causes the DB\_datasets RPC to set the max\_records field to the maximum number of records that the data set can hold without expanding. This is quite useful in estimating the size of a table. In the case of variable format data sets, the number is meaningless, as it uses the size of the fixed part to compute the value of max\_records.

HOST ERROR: Databridge call failed for sets, error number = *number*

Databridge was unable to retrieve information on sets. This message can occur when you run a `define` or `redefine` command.

HOST ERROR: Databridge call failed for switch\_audit, error number = *number*

The Client remote procedure call on the DBSWITCHAUDIT API failed. This message can occur during a `switchaudit` command.

## Sockets Errors

Sockets errors appear onscreen and in the log file. If tracing is enabled, messages are also captured in the trace file.

Sockets errors typically indicate the following:

- ◆ LAN problems
- ◆ Host TCP/IP problems (for example, you have reached the maximum TCP/IP session quota)

SOCKETS ERROR: connect call failed, error=*number*(*errortext*)

The Client is unable to connect to the Databridge server. In this case, check the following:

- ◆ Did you enter the correct host name or IP address?
- ◆ Did you enter the correct port number? The port number you enter must match the port number as it is defined in DBServer parameter file.
- ◆ Is the host running? If so, is Databridge Engine and DBServer running?
- ◆ If you use DBEnterprise Server, is the DBDirector service running?

SOCKETS ERROR: getaddrinfo call failed, error=*number* (*errortext*)

This message appears when the Client cannot resolve any of the host names supplied to the Client via the `define` command. Check the following:

- ◆ Did you enter the correct host name?
- ◆ If you entered the host name, did you set up your domain name resolution parameters correctly in the Windows Network folder (in control panel)?
- ◆ If you entered the host name, is the domain name server running? If the domain name server is down, change the host name in the DATASOURCES table to the IP address and try the command again.

SOCKETS ERROR: gethostname call failed, error=*number* (*errortext*)

This message when DBClient or DBCIntCfgServer tries to get the host name when trying to connect to the service. This is an internal that should never occur. If it does contact Micro Focus support.

SOCKETS ERROR: recv call detected a closed {server | console} connection

This message appears when the Client is reading TCP data, and indicates that the Client received an abnormal status that indicates the connection to DBServer is closed.

```
SOCKETS ERROR: recv call failed for {server | console} connection, error=number (errortext)
```

This message can occur during a `define`, `process`, `clone`, or `switchaudit` command. It indicates that the Client encountered an error while receiving data. This error typically occurs when DBServer aborts or resets the TCP/IP connection. The `errortext` that is provided should help clarify what happened.

```
SOCKETS ERROR: recv call returned 0 after select reported data available
```

This message, which only occurs when the connection to DBServer is being initialized, indicates that the Client could not obtain the requested data. This occurs only when DBServer closes the connection.

```
SOCKETS ERROR: select timed out after sss seconds for server connection
```

This message appears when the Client is starting up a connection to the Databridge server. We use an echo RPC to determine if the line is operational or not. We use a sockets select call to timeout the receipt of the response to the ATM\_Echo RPC. When you get no data after the prescribed timeout period you will see this message. Normally, the client does not immediately give up, it tries to recover from this error.

```
SOCKETS ERROR: send call failed for {server | console} connection, error=number (errortext)
```

This message can occur during a `define`, `process`, `clone`, or `switchaudit` command. It indicates that the Client encountered an error while sending data. This error typically occurs when DBServer aborts or resets the TCP/IP connection. The `errortext` that is provided should help clarify what happened.

```
SOCKETS ERROR: setsockopt(SO_KEEPALIVE) call failed for TCP port, error=number (errortext)
```

This message, which can occur during any command that involves a TCP/IP connection, indicates that the Client was unable to set the keep-alive option for the socket. When the connection is idle, the keep-alive option prompts a message to be sent so that the Client can determine if the connection is still alive.

```
SOCKETS ERROR: socket call failed, error=number (errortext)
```

This message can occur at the beginning of a `define`, `process`, `clone`, or `switchaudit` command. It indicates a problem in the Windows sockets software. The `errortext` that is provided should help clarify what happened.

```
SOCKETS ERROR: WSStartup error ddd
```

This message applies only to the Windows Client. This message can occur at the beginning of a `define`, `process`, `clone`, or `switchaudit` command. It indicates that the Windows sockets software is incompatible with that used by the Client. In your Winsock documentation, look up the error code indicated by `ddd`. If you cannot resolve the problem, contact Micro Focus Technical Support.

## ODBC Errors

All ODBC calls return a result code. The most common result codes are the following:

SQL\_SUCCESS  
SQL\_SUCCESS\_WITH\_INFO  
SQL\_ERROR

SQL\_SUCCESS indicates that the call was successful. SQL\_SUCCESS\_WITH\_INFO indicates that the call completed successfully, possibly with a nonfatal error (warning). The Client will sometimes suppress such warnings if they are expected; in all other cases, these warnings are displayed.

SQL\_ERROR indicates that the call encountered an error. In case of errors, the Client will interrogate ODBC to get the details of the error it displays.

Diagnostic messages may include several error messages. Each of these errors includes an `SQLSTATE` that provides detailed information about the cause of a warning or error and a diagnostic message that includes a native error code (generated by SQL Server). An example of such an error is:

```
RetCode = SQL_SUCCESS_WITH_INFO, SQLState = 01000; native_error = 5701,  
error = [Microsoft][SQL Native Client][SQL Server]Changed database context  
to 'TESTDB'.
```

This message is generated when the Client connects to the database. The native error of 5701 is generated by SQL Server.

The following example illustrates an error that typically occurs during the fixup phase. The client suppresses such messages during the fixup phase, but displays them during the change tracking phase.

```
RetCode = SQL_ERROR, SQLState = 23000; native_error = 2601, error =  
[Microsoft][ODBC SQL Server Driver][SQL Server]Cannot insert duplicate key  
row in object 'customer' with unique index 'customer_set'.
```

In some situations you may get multiple error messages caused by the same event. These kinds of error messages are generated at different levels of the ODBC interface. The Client uses the `native_error` code in all cases that involve errors generated by the database.

## OCI Errors

OCI errors appear onscreen and in the log file. If tracing is enabled, messages are also captured in the trace file.

OCI errors are prefixed by the time stamp `hh:mm:ss`, where `hh` is hours, `mm` is minutes, and `ss` is seconds. In addition, each of the OCI errors is followed by the error message text in the following form:

```
Error Text: text
```

where `text` is the error text returned by the OCI. The text may sometimes be several lines long, especially if more than one error has occurred. The text starts with `ORA-xxx`, where `xxx` is the Oracle error number. Explanations are in your Oracle documentation.

OCI errors typically precede a Databridge error. You can typically find the source of the problem by looking up the OCI message. If the error involves a stored procedure, you must search through line numbers and stored procedure names to find the error.

The following OCI error is most likely to occur with the Databridge Client for Oracle.

OCI ERROR: *err\_code = nmmn for ocifunction*

where *ocifunction* is the name of the OCI function involved in the error. The following is a list of possible OCI functions.

OCIAttrGet  
OCIAttrSet  
OCIBreak  
OCIDefineByPos  
OCIEnvCreate  
OCIEnvInit  
OCIHandleAlloc  
OCIHandleFree  
OCIServerAttach  
OCIServerDetach  
OCISessionBegin  
OCISessionEnd  
OCISstmtExecute  
OCISstmtFetch  
OCISstmtPrepare  
OCITransCommit  
OCITransRollback

For more details, see your Oracle documentation.

## Client Warnings

Warning messages appear onscreen and in the log file. If tracing is enabled, messages are also captured in the trace file. These messages are prefixed by the timestamp *hh:mm:ss* (where *hh* is hours, *mm* is minutes, and *ss* is seconds).

WARNING: "-t 1" option ignored as this would create a second copy of the log file

This message indicates that for the Client to process this option, additional bits must be set in the trace mask to differentiate the resulting trace file from the log file.

WARNING: AA values for DataSet *name[/rectype]* are no longer valid, *ds\_mode* set to 0 to force a re-clone

This message, which occurs during a *redefine* command or when running the Client Configurator, indicates that the specified data set must be re-cloned because its AA Values are no longer valid after a DMSII reorganization.

WARNING: AA values for DataSet *name[/rectype]* no longer valid

This message, which can occur during a *process* command, indicates that a DMSII structural reorganization has invalidated the specified data set that uses AA values as a key. The program automatically marks the data set as no longer trackable unless they are re-cloned (that is, sets the *ds\_mode* to 12). The *redefine* command will take corrective action and set the data set's *ds\_mode* to 0. If your data sets have links, avoid using the *redefine* command to achieve this end, as it may not work correctly. Adding the *-y* option on the *process* command is a much better way to force the data sets to be re-cloned.

WARNING: AA values for DataSet *name[/rectype]* are no longer valid after reorg

This message, which can occur during a `redefine` command, indicates that a DMSII structural reorganization has invalidated the keys for the specified data set, marking the data set as needing to be recloned (`ds_mode` to 0). The specified data set uses AA values as a key.

WARNING: Adding a UNIQUEIDENTIFIER column to a table is not supported -- you need to reclone

This message appears when a `redefine` encounters a new column that is being stored as a **uniqueidentifier**. Since all existing rows would have their own GUIDs you must reclone this table.

WARNING: Adding AA values to a table is not supported -- you need to reclone

This message, which can only occur during a `redefine` command or when running the Client Configurator, indicates that the command detected that the table uses AA values, which were previously not used. Since we cannot assign the correct values for this column in the existing records, you must reclone the data set.

WARNING: AFN *nnn* has no StateInfo updates

This message appears when an audit file that contains no quiet points is processed. This means that you either have a very long transaction or that the audit file is empty.

WARNING: All database API functions disabled by 'z' option

This message occurs during a `process` or `clone` command with the `-z` option. The Client loads the Client control table entries for the selected data source, after which it connects to the host and sets a global flag that disables all relational database functions.

The `-z` option is a debugging tool that can indicate the non-database time required to extract data for a data set.

WARNING: All Engine COMMIT parameters are zero; this needs to be corrected immediately as it will degrade performance immensely

This message, which can occur after an operator console command that changes a commit parameter, indicates that the resulting commit parameters are in error, as they do not specify any situation under which the Engine should try to do a commit. Since this was caused by an operator command, the client does not exit, but instead tells the operator that the actions taken need to be reversed.

WARNING: All members of the variable format DataSet *name* must be recloned, `ds_mode` set to 0 for retype *nnn*

If the `ds_options` bit `DSOPT_Split_Vfmt_ds` is set or the configuration parameter `split_varfmt_ds` is enabled, the Databridge Client won't support the cloning of only certain members of variable format data sets.

WARNING: Archive file was created by a newer version of the program, extra fields will be ignored

This message can occur during a `reload` command when you attempt to read an archive file that has a higher version than what the program supports. There is no guarantee that this will work, as the program just reads the number of fields it knows about and ignores the rest. We do not recommend trying to read archive files that were created by newer Clients. Use the `-V` option to write the archive files in formats compatible with older clients. For a list of versions, see "dbutility Command-Line Options" in the *Databridge Client Administrator's Guide*.

WARNING: Attempt to clear duplicate records for table '*name*' failed

During the data extraction, it is possible to get duplicate records. When a record is moved, it appears more than once in the data set. If the `ds_options` bit `DSOPT_ClrDup_Recs` (32768) is set in the `DATASETS` table entry for the data set, the Client remedies this by running the script `script.clrduprecs.tabname`. This script deletes all records that appear multiple times in the table. The fixup phase involves the processing of audit files that span the times during which the data extraction was done for the data set, then reinstates the correct copy of these records. If the script fails, this error results. See the SQL errors that precede this message to determine why the script failed.

Keys that allow nulls may cause problems with this script and require that you manually remove the duplicate records and recreate the index. Then, you can set the `ds_mode` for the data set back to 1 and restart the Client.

WARNING: Attempt to drop procedure `U_name` failed

This message, which only occurs during a `configure` or `dropall` command, indicates that the Client was unable to drop the specified stored procedure (`U_name`, where `name` refers to a control table name). Refer to the database API error messages that precede this message to determine the cause of this error. This error will occur if your `userid` has inadequate privileges for dropping a stored procedure. Starting with version 6.5, the Databridge Client no longer uses stored procedures to update control tables, it uses the actual SQL statement in conjunction with host variables instead.

WARNING: Attempt to modify record in table '`name`' after failed insert was unsuccessful, changing it to a delete/insert (`AFN=afn`, `ABSN=absn`, `SEG=seg`, `INX=inx`) - Keys: `columnname = value`, ...

This message, which can occur during a `process` or `clone` command, indicates that the recovery for a failed insert failed. Normally, the Client tries to recover by doing an update instead of an insert. When the update fails, the Client tries to recover by doing a delete/insert operation. If the specified table is an `OCCURS` table, all of the rows for the given record are deleted and reinserted. Under normal circumstances, the Client should recover from this situation. This warning indicates that the table likely has missing rows or extra rows, either of which are causing the update to fail.

WARNING: Attempt to propagate global stateinfo for DataSets failed

This message can occur during a `process` or a `clone` command after an audit file switch, or when the Client shuts down at the end of the command. It indicates that the program's attempt to propagate the global state information to data sets with their `in_sync` columns set to 1 failed.

WARNING: Attempt to run back-to-back `redefine` commands; control tables will be reloaded from the backup file

This message indicates that you are running back-to-back `redefine` commands, which can result in incorrect `redefine` scripts being generated. To prevent this from happening the client reloads the control tables from the unload file created by the first `redefine` command, thus eliminating the possibility of creating bad `reorg` scripts.

WARNING: Attempt to set MARS failed -- reverting to single thread operations

When multi-threaded updates are enabled, the SQL Server Client enables MARS (Multiple Active Result Sets), a Microsoft ODBC extension option. MARS allows multiple queries that get results to be executed simultaneously. This is done by a `SQLSetConnectAttr` ODBC call. If this ODBC call gets an `IM006` error indicating that the operation failed, this warning is issued and multi-threaded updates are disabled. Make sure that you are not using an old SQL Server ODBC driver.

WARNING: Attempt to set parameter `param_name` to `value1` rejected by DBEngine; using `value2` instead



This message indicates that Databridge Engine did not set a parameter value as specified by the Client because it was not within the configured allowable range. To change the configured allowable range, contact the administrator for the Databridge host software on the mainframe.

WARNING: Attempt to update {dms\_scale | dms\_signed} ignored, this is only valid for an ALPHA item being cloned as a NUMBER

When cloning an ALPHA item as a number, you can set the dms\_scale and dms\_signed columns in DMS\_ITEMS to indicate that the item has a scale and/or is signed. This error indicates that the attempt to perform these updates from the Console failed. Look for the corresponding SQL error in the client log file to determine why the operation failed.

WARNING: Attempt to update stateinfo for DataSets in tracking mode failed

The attempt to update the stateinfo in the DATASETS table failed from the Client Console, see the SQL errors that precede this message to determine why the operation failed.

WARNING: Bad DMS Timestamp 0xhhhhhhh received, Timestamp set to NULL

This message can occur during a process or clone command when processing a stateinfo record that has a bad time stamp. This indicates an internal error from which the program recovers by setting the timestamp to NULL to prevent a SQL error.

WARNING: Bad DMS Timestamp 0xhhhhhhhhhhhh received for item 'name' in table 'name', {record will be discarded | date set to NULL} - Keys: columnname = value

This message can also occur when attempting to interpret a DMSII REAL or ALPHA(6) item as a TIME(6) value and the value of the item is illegal. This can be caused by one of the following problems:

- ◆ Bad DMSII data
- ◆ Attempting to interpret data that is not a DMSII TIME(6) as a TIME(6)
- ◆ The Client recovers by setting the value of the item to a NULL, unless it is a key, in which case the record is discarded.

WARNING: Bad DMS TIME(60) value 0xhhhhhhhhhhhh received for item 'name' in table 'name', {record will be discarded | date set to NULL} - Keys: columnname = value

This message can appear during a process or clone command. It indicates that the value of a DMSII REAL or ALPHA(6) item, which is being interpreted as a TIME(60) value, is illegal. This can be caused by one of the following problems:

- ◆ Bad DMSII data
- ◆ Attempting to interpret data that is not a DMSII TIME(60) as a TIME(60)
- ◆ The Client recovers by setting the value of the item to a NULL.

WARNING: Bulk copy record count (mmm) and actual record count (nnn) differ for table 'name'

This message, which only appears if the configuration parameter verify\_bulk\_load is set to 1, indicates a mismatch between the number of rows in the table and the number of rows that the Client loaded. Check the bulk loader log file and the discards directory for a discard file for the given table. If the parameter verify\_bulk\_load is set to 1, the Client ignores the error. However, if it is set to 2, you get a similar error and the Client abends.

WARNING: Bulk load failure has occurred, continuing with process command

When the Client detects a bulk load failure, it adds the data set to a list of data sets that encountered bulk loader failures during the data extraction phase and then it prints this message.

When the data extraction phase ends, the Client displays a message for each data set in the list. This message is the initial indication of the failure, which is reported again when the data extraction completes. Note that if the `verify_bulk_load` parameter is set to 2, the client aborts as soon as it gets the first bulk loader failure.

WARNING: Bulk loader discard file created for table '*name*'

This message indicates that the program detected the existence of a bulk loader discard file (`sqlld.name.bad` or `bcp.name.bad`) after table loading was complete. This is an indication that the bulk loader was not able to load all of the records. Look at the bulk loader discard and log files to determine why these errors occurred.

WARNING: Bulk loader thread appears to be hung; main thread waiting

(Windows only) This message indicates that the bulk loader thread could not continue after passing a temporary data file to the bulk loader. Typically, this occurs because the amount of temporary data file storage exceeds the value of `max_temp_storage`. This message indicates that the bulk loader thread is not finished loading the previous batch of files. If this is caused by the bulk loader simply being slow, the situation will correct itself. If the bulk loader thread has hung, look at the `sql*loader` or `bcp` log files to determine the cause.

WARNING: Cannot drop primary key or index for table '*name*'

This message indicates that the Client encountered a SQL error while trying to drop an index. The index was being dropped because a bad value in `dt_options` of the `DATATABLES` table caused the index type to be wrong. This error can occur if you change the index types that the Databridge Client uses by using means external to the Client (not recommended). See the SQL errors that precede this message to determine why the operation failed.

WARNING: Cannot get Oracle version

This message can occur during any Client command. It indicates that the Client was unable to read the Oracle system table that contains the product version. Contact Micro Focus Technical Support.

WARNING: Cannot get SQL Server version

This message can occur during any Client command. It indicates that the Client was unable to read the SQL Server version using a "select @@version" SQL statement. Contact Micro Focus Technical Support.

WARNING: Cannot reorganize table '*name*'

This message can occur during a `redefine` command or when running the Client Configurator. It is usually preceded by other messages that indicate why the table cannot be reorganized. In this case, the Client sets the `ds_mode` column to 0 in the corresponding entry in `DATASETS` table to force the data set in question to be reloaded.

WARNING: Client operations inhibited between *hh:mm* and *hh:mm*

This message indicates that the Client cannot execute a `process` or `clone` command because a blackout period is in effect. Blackout periods are defined by the `blackout_period` parameter in the scheduling section of the Client configuration file.

WARNING: Clone of DataSet *name*[/*rectype*] failed during bulk load

When the data extraction phase ends, the Client displays this message for all data sets that encountered bulk loader failures. The actual failure will have already been reported as a bulk loader error. This is done so that you don't have to look through the log file to find the data sets in question.

WARNING: Clone of DataSet *name*[/*rectype*] was not completed

When Enterprise Server encounters a situation that forces it to restart the clone using remote regions (also referred to as "indirect disk"), it aborts the clone. The Client keeps track of this fact for the affected data sets. After the data extraction phase is completed, if any of these data sets have a mode of 0 (that is, the clone was not successfully completed), the Client displays this message.

WARNING: Column name '*name*' already in use, using '*name*' instead

This message can occur during a `define` or `redefine` command. It indicates that a duplicate column name was changed to the specified name to make it unique.

WARNING: Command will have no effect until the next scheduled execution time

This message indicates that the user entered a `SCHED` command from the `dbutility` console while the Client was idle.

WARNING: Control tables do not exist, there is nothing to do

This message can occur during a `dropall` command and indicates that there is nothing to drop because the Client control tables do not exist.

WARNING: Data Source *name* has INDEPENDENTTRANS set, `extract_embedded` parameter ignored

This message indicates that `INDEPENDENTTRANS` is set to `True` and therefore, your setting for the `extract_embedded` parameter is disregarded. If `INDEPENDENTTRANS` is `False`, and `extract_embedded` parameter is `True`, Databridge allows the data extraction of embedded data sets, even though these data sets cannot be tracked.

WARNING: Database API functions previously disabled by '`z`' option are now enabled

This message occurs when a `process` or a `clone` command is run with the `-z` option and indicates that the `-z` option is no longer in effect. This is just a reminder that the relational database did not get updated by this run.

WARNING: Database update level change ignored

This message can occur during a `process` or `clone` command and indicates that the Client ignores the change in the DMSII update level because none of the selected data sets were affected by the DMSII reorganization. You should only see this message if you are using a `MISER` database, in which case the `miser_database` parameter is set to `True`. We require that you run a `redefine` command regardless of whether the Engine thinks that no data sets are affected. The Engine does not recognize changes to a `SET` that the Client might be using as the source for the index. We recommend that you set the `auto_redefine` option when using the service, as this allows the service to automatically run a `redefine` command and prevent interruptions to the Databridge operations.

WARNING: DataSet *name*[/*rectype*] allows duplicates, no updates will be allowed

This message indicates that the specified DMSII data set does not have any SETS that do not allow duplicate records and that the AA Values cannot be used as keys. The data sets will be cloned; however, subsequent runs of `process` commands will not track changes for the data set. The only time you receive this message is when you run a `generate` command. The solution is to use an RSN as a key, as explained in Chapter 2 of the *Databridge Client Administrator's Guide*.

WARNING: DataSet *name*[/*rectype*] can be reorganized, run `reorg` command: `ds_mode set to 31`

This message, which occurs when running a `redefine` command or the Client Configurator, indicates that the program has generated scripts to reorganize the tables for the specified data set and avoid recloning it.

Make sure that you examine the scripts for problems and back up your relational database before you run a `reorg` command that will alter the tables.

WARNING: DataSet *name* does not exist -- reload will not be performed

This message can occur during a `dbutility reload` command, if a list of data sets is specified on the command line and one of them is found not to exist. This message is just a reminder that the data set name in question was ignored.

WARNING: DataSet *name*[/*rectype*] does not have valid AA values, `Include_AA` option ignored

This message can occur during a `define` or `redefine` command and indicates that the Client ignores the `DSOPT_Include_AA` option, because the specified data set does not have valid AA values. You should not use the `ds_options` bit `DSOPT_Include_AA` (16), as this bit is deprecated. Use the `ds_options` bit `DSOPT_Use_AA_Only` (0x4000) to force the Client to use the AA Values (or RSNs) as the source for the index.

WARNING: DataSet *name*[/*rectype*] does not have valid AA values, `Use_AA_Only` option ignored

This message, which occurs during a `define` or `redefine` command, indicates that the Client is ignoring the `DSOPT_Use_AA_Only` option, because the specified data set does not have valid AA values.

WARNING: DataSet *name*[/*rectype*] has been added

During a `process` command, the Client compares the new data set list from the Engine to the old data set list when a DMSII reorganization has been detected. If the new data set list contains additional data sets, this message is displayed. The ensuing `redefine` command will pick up these data sets unless the configuration parameter `suppress_new_datasets` is set to True.

WARNING: DataSet *name*[/*rectype*] has been deleted

During a `process` command, the Client compares the new data set list from the Engine to the old data set list when a DMSII reorganization has been detected. If the new data set list is missing any data sets, this message is displayed.

WARNING: DataSet *name*[/*rectype*] has been initialized

This message can occur during a `process` or `clone` command and indicates that the specified data set has been initialized in the DMSII database (that is, all records deleted from it).

WARNING: DataSet *name*[/*rectype*] has been reorganized

This message can occur during a `process` or `clone` command if the Databridge Engine detects a structural reorganization or a change in item counts (for example, a filler substitution). For every data set affected by the reorganization, the Databridge Engine sends the Client a stateinfo record whose `ds_mode` field has a value of 3. The Client sets the `DS_Needs_Redefining` bit in the `status_bits` column of the corresponding DATASETS entry and displays this message.

WARNING: DataSet *name[/rectype]* has been reorganized, you must first execute a `redefine` command

This message can appear during a `process` or `clone` command if you try to select a data set whose format level is not current. The Client sets the `DS_Needs_Redefining` bit in the `status_bits` column of the DATASETS entry for the data set. It also sets a flag that causes the Client to exit with an exit status value of 2 immediately after all the data sets have been selected. It does not stop immediately so that it can catch all such data sets before you run a `redefine` command. The `redefine` command will most likely require that you run a `reorg` or `generate` command before resuming operations.

WARNING: DataSet *name[/rectype]* has been reorganized (fmtlevel: *mmm* -> *nnn*)

During a `process` command, the Client compares the new data set list from the Databridge Engine to the old data set list when a DMSII reorganization has been detected. This message only appears when the Databridge Engine fails to detect the fact that a data set has been reorganized. The Client handles the reorganized data set accordingly.

WARNING: DataSet *name[/rectype]* has `ds_mode=0`, run a `reorg` (or `generate`) command before re-cloning it

This message, which occurs when running a `redefine` command or the Client Configurator, indicates that the specified data set will be re-cloned and program scripts must be generated (that is, via the `generate` command). If other data sets are involved, the `generate` command is automatically invoked when you run a `reorg` command. If you do not run a `reorg` command, you must execute a `generate` command before you reclone the data set.

WARNING: DataSet *name[/rectype]* has `ds_mode=mm`, mode set to 0

This message occurs during a `redefine` command or when running the Client Configurator. It indicates that a data set has an invalid `ds_mode` and will be re-cloned as a result. Note that, a `ds_mode` value of 99 is used by the Client to indicate bulk loader failure.

WARNING: DataSet *name[/rectype]* has embedded datasets, AA values will be used as keys

This message can occur during a `define` or `redefine` command. It indicates that the Client will use AA values (or RSNs) as the keys for the specified data set.

WARNING: DataSet *name[/rectype]* has invalid AA\_Values, it will not be selected

This message, which occurs during a `process` or `clone` command when data sets are selected, indicates that the Client encountered a data set whose `ds_mode` is 12 and whose active column is 1. The `ds_mode` value of 12 indicates that the AA values for the data set are no longer valid. The Client treats this type of data set as if its active column was 0 until it is re-cloned. To force a re-clone, you must set `ds_mode` to 0 using the Databridge Client Console or the relational database SQL query tool, or run the `process` command with the `-y` command-line switch.

WARNING: DataSet *name[/rectype]* has no active tables; active column set to 0

This message, which can occur during a `process` or `clone` command, indicates that the active columns have been set to 0 in all tables associated with the specified data set. In this case, the Client automatically sets the data set's active column to 0, as there are no tables to populate.

WARNING: DataSet *name[/rectype]* has no index, it will not be selected

This message can occur during a `process` or `clone` command and indicates that the specified data set will not be selected because its `ds_mode` had a value of 11, which indicates that it has no index and therefore cannot be tracked.

WARNING: DataSet *name[/rectype]* has no unique keys, AA values will be used as keys

This message can occur during a `define` or `redefine` command. It indicates that the Client will use AA values (or RSNs) as the keys for the specified data set.

WARNING: DataSet *name[/rectype]* has no unique keys and AA values are not usable as keys - Change tracking is not possible unless a composite key is created

This message can occur during a `define` or `redefine` command. It indicates that the specified data set has no index and that AA values cannot be used as the keys. Your only recourse is to create a composite key or possibly enable RSN for the data set in DMSII (EXTENDED attribute).

WARNING: DataSet *name1[/rectype]* is a remap of DataSet *name2[/rectype]*, active column set to 0

The term remap is a DMSII term that is roughly equivalent to the term view in a relational database. In DMSII, a data set can have one or more remaps. The original data set is referred to as the base structure. It is also possible to have a remap for a variable-format data set. Note that */rectype* only appears for variable-format data set records that have a nonzero record type (that is, contain a variable part). By default, the Client only clones the base data set and sets the active column to 0 for all remaps. If you want to use a remap instead of the base structure, you can do so by changing the values in the active columns in the DATASETS table for the data sets in question.

WARNING: DataSet *name[/rectype]* is embedded and INDEPENDENTTRANS is reset - Change tracking is not possible

When INDEPENDENTTRANS is set to FALSE in the DASDL, the parent-child relationship for embedded data sets is lost, and it's impossible to track changes for these data sets. You can clone them if you set the configuration parameter `extract_embedded` to True.

WARNING: DataSet *name[/rectype]* is not the first remap of structure #*strnum*, active column set to 0

This warning, which can occur during a `define` or `redefine` command, indicates that the search for the base structure of a remap was unsuccessful, but we found more than one remap of the base structure and are using the first remap as a substitute for the base structure. If this is not what you want, you can rectify the situation by changing the active columns of the remaps of the base structure so that the client picks the one you want.

WARNING: DataSet *name[/rectype]* must be re-cloned, run `reorg` (or `generate`) command: `ds_mode` set to 0

This message occurs during a `redefine` command or when running the Client Configurator. It indicates that the data set in question must be re-cloned because the layout changes cannot be applied using `alter` commands. To resolve the issue, the scripts for the table need to be recreated by running either a `reorg` or a `generate` command. (The `reorg` command executes a `generate` command.)

WARNING: DataSet *name[/rectype]* record size sent by the Databridge Engine is too short, length changed from 0xhhhhh to 0xhhhhh)

The Client computes the sizes of data set records and compares its result to those passed to it by the Engine in the *recsz\_bytes* column of the DATASETS table. If the resulting length is larger than the one provided by the Engine the Client uses the computed value and issues this warning to make it known that the Engine is sending the wrong length. Notify Micro Focus Technical Support if you ever get this message.

WARNING: DataSet *name[/rectype]* which contains embedded DataSets does not have valid AA values

This message can occur during a *define* or *redefine* command and indicates that the specified data set contains embedded data sets but no valid AA values.

WARNING: DataSet *name[/rectype]* which contains links does not have valid AA values

This message can occur during a *define* or *redefine* command and indicates that the specified data set contains link items but no valid AA values. These types of links cannot be supported.

WARNING: DataSet *name[/rectype]* will be recloned because DataSet *name1[/rectype1]* is being recloned

The *process* command will reclone the specified data set because the following conditions exist:

The configuration parameter *automate\_virtuals* is set to True.

The specified data set is the secondary source for a virtual data set that gets its input from more than one DMSII data set. This data set was cloned previously but does not have a *ds\_mode* value of 0.

The data set that is the primary source for the virtual data set has a *ds\_mode* of 0 (it is being recloned).

For example, assume that SV-HISTORY and SAVINGS both provide data for the virtual data set SV-HISTORY-REMAP, and SV-HISTORY is the primary source. When the *process* command finds that SV-HISTORY is being recloned, it ensures that SAVINGS is also recloned. Unlike the *clone* command, which produces an error in this case, the *process* command simply reclones the specified data set after issuing this warning.

WARNING: DataSet *name[/rectype]* will be treated as if its active column was 0 - execute a *redefine* command to remove its definitions from the client control tables

This message is displayed during a *process* or *clone* command when the program selects the data set and gets an invalid structure number error for DBEngine. This can happen at sites that create new data set structures that replace old ones with the same name. To avoid this problem, run a *redefine* command followed by a *generate* command to get the new structure defined. Then, reclone the data set in question. To avoid recloning, you must record the old stateinfo and reinstate it to get the program to produce exactly the same layout for the tables.

WARNING: DBEnterprise audit files origin changed from *AF\_origin1* to *AF\_origin2*

The Client displays this message when the audit file origin for Databridge Enterprise Server changes. This message is not displayed if the previous value of the audit file origin is unknown (255).

WARNING: DBEnterprise audit files origin is *AF\_origin*

The Client displays this message when the audit file origin for Enterprise Server changes from unknown (255) to a value different than that specified by the configuration parameter `dbe_dflt_origin`. For example, the parameter is set to `DirectDisk`, but the new value is `IndirectDisk`.

WARNING: Deleting column '*name*', which is a key, requires recloning DataSet *name*[/*rectype*]

This message, which can occur during a `redefine` command, indicates that a column that used to be a key is no longer present. When this happens the client will want to reclone the data set, as failure to do so could result in duplicate records. One such example would be the deletion of an external column of type `deleted_record`.

WARNING: "`discard_data_errors`" parameter overrides "`alpha_error_cutoff`" parameter

This message can appear during any Client command, it indicates that configuration file specifies two mutually exclusive configuration parameter settings. The `discard_data_errors` parameter forces the `alpha_error_cutoff` parameter to be set to 0, indicating that even if one character is in error, the Client should treat the data as bad and discard the record.

WARNING: DMSII reorganization has occurred; AA values for DataSet *name*[/*rectype*] no longer valid - {add the `-y` option to the process command or set `ds_mode` to 0 to force a reclone}

This message indicates that a DMSII garbage collection reorganization has invalidated the keys of the specified data set that uses AA values as keys. You must reclone the affected data set. The Databridge Client sets the `ds_mode` column of such a data set to 12.

The Client deselects the specified data set and continues the command. The next time you run a `process` command, the Client will not select data sets whose mode is 12. You can force a reclone by specifying the `-y` switch for the next `process` command. Do not set `ds_mode` to 2 unless you create a composite key instead of using the column `my_aa`. See "Changing an Index from AA Values to a Composite Key" in Chapter 4 of the *Databridge Client Administrator's Guide*.

WARNING: `DTOPT_Preserve_Deletes` bit erroneously set for table '*name*' -- option reset

This message indicates that the program encountered a table whose `DTOPT_Preserve_Deletes` bit is set in the `dt_options` column of the `DATATABLES` entry even though there are no user columns that are required to implement deleted record preservation. Make sure that you did not inadvertently set this bit.

WARNING: `DTOPT_PrimaryKey` option does not match index type for table '*name*'; attempting to recover

This error message can occur during a `reorg` command when the client tries to drop an index. If the `dt_options` bit for the table is wrong, the client realizes this and recovers by using the right SQL to drop the index.

WARNING: Duplicate deleted record for table '*name*'; use delete sequence number to avoid slowing the program down



This message can occur when you are preserving deleted records using the `deleted_record` column. Since we use the current value of time as the value in this column, when more than one deletion of the same record is seen by the Client within the same second, a duplicate record is created. The client has a very inefficient way of dealing with this situation, which is to sleep for a second waiting for the value of time to change. To avoid this situation you need to add the `delete_seqno` column, which eliminates such duplicates. Under normal circumstance this situation should not occur, unless you have programs that do updates by doing delete/insert operations rather than actual updates. It is therefore recommended that you always add the sequence number column, when preserving deleted records.

WARNING: Duplicate record encountered during a two step modify for table '*name*' which has unique keys - Program automatically recovers from this error by doing a delete/insert instead (AFN=*afn*, ABSN=*absn*, SEQ=*seg*, INX=*inx*)- Keys: *columnname* = *value*, ...

This warning, which can occur during a `process` or `clone` command, indicates that the Client was unable to update a record in the specified table during a two-step modify. A two-step modify occurs when a change occurs to the value of the depends item for an item with an OCCURS DEPENDING ON clause. The Client first updates the rows that remain in the OCCURS table and later deletes the rows that are no longer present, or adds rows that were inserted. This warning appears if the update for one of the rows fails because the target row is not in the table. The Client recovers by deleting all of the rows in the OCCURS table for the given key and inserting all of the rows that are present in the after image. This only happens when the OCCURS table is corrupt.

WARNING: Duplicate record encountered for table '*name*' which has unique keys - Program automatically recovers from this error by doing an update instead (AFN=*afn*, ABSN=*absn*, SEQ=*seg*, INX=*inx*) - Keys: *columnname* = *value*, ...

This message can occur during the `fixup` phase of a `process` or `clone` command if the `-v` option is enabled. It is unconditionally displayed if this situation occurs during the update phase as this is not expected behavior. The Client attempts to deal with this situation by changing the failed insert into an update instead.

WARNING: Duplicate record encountered for table '*name*' which has unique keys - Program automatically recovers from this error by doing a delete/insert instead (AFN=*afn*, ABSN=*absn*, SEQ=*seg*, INX=*inx*) - Keys: *columnname* = *value*, ...

This message can occur during when an update finds no rows to update and the subsequent insert get a duplicate message error. As a last ditch effort the client tries to delete the record and reinsert it. Sometimes the presence of triggers can adversely affect the row count, which is about the only way to explain how this situation could ever occur. If the delete/insert works, the client continues processing updates, thus avoiding an unnecessary stop.

WARNING: End\_Split\_Table bit in `di_options2` for item name in DataSet name[*rectype*] has no matching item with the Split\_Table bit set; the bit will be ignored

The bits `DIOPT_Split_Table` (4) and `DIOPT_End_Split_Table` (8) in the `DMS_ITEMS di_options2` column allow you to move a block of DMS items to a new table and return back to adding items to the original table. The current implementation does not allow consecutive `DIOPT_Split_Table` bit setting, as we only remember the last split where the table was split. This warning indicates that you have a spurious `DIOPT_End_Split_Table` bit set due to a bad user script.

WARNING: Error in creating index for FileExtract table '*name*' was ignored

Unlike DMSII data sets, FileXtract tables do not require an index for data tracking. The only type of update that is performed is an insert. Under these conditions, the Client does not object to the lack of an index. This message simply documents the fact that there was no index created.

WARNING: Error in creating index for table '*name*'

This message indicates that an error occurred when creating an index for the specified table. This can happen if:

- ◆ You have a very large table and your log space is too small.
- ◆ You have duplicates in a table that is defined with a unique key.
- ◆ You created an invalid composite key.

When this error occurs and the DATASETS *ds\_options* bit *DSOPT\_ClrDuprecs* (0x8000) is set to 1, the Client runs the script *script.clrduprecs.name*. If this script runs successfully, the index is re-created. If the bit is not set or the index creation fails after the script runs, the Client sets *ds\_mode* to 11 to indicate that the data set has at least one table that does not have an index. This puts the data set temporarily out of commission until you fix the problem. To create the index for the specified table, issue a *runscript* command for *script.index.tablename*, as follows:

```
dbutility -n runscript dbscripts\script.index.tablename
```

WARNING: Error in dropping DataSource *name*

An error occurred during a *dbutility dropall* command. Check the previous error messages (on screen or in log file) for details on what went wrong. If you attempted a *dropall* command and the scripts for the specified datasource could not be found, you probably did not point your current directory to the proper place. If this is the case, change to the proper directory and repeat the *dropall* command.

This error does not stop the execution of the *dropall* command. Instead, the Client informs you via this message and then proceeds to the next data source. When the *dbutility dropall* command is finished with all of the data sources, it leaves the Client control tables. This allows you to fix the problem (for example, the scripts are in a different directory). You can then rerun the *dropall* command.

WARNING: Error in initializing table '*name*'

The specified table was not dropped and recreated successfully. Check the error messages that appear prior to this one for more details.

WARNING: Error in reading Null Record file "*datasource\_NullRec.dat*"

This message can occur during a *process* or *clone* command and indicates that the Client encountered an error while reading the specified file. Check the error messages that precede this message to see why this file could not be read.

WARNING: Error in reading value in BI for depends item '*name*' in table '*name*', using maximum of *nnnn* instead

This warning, which can occur during a *process* or a *clone* command, indicates that the Client detected a data error while retrieving the value of the depends item from the before-image of the record. A depends item is an item pointed to by an OCCURS DEPENDING ON clause, which is in the form "OCCURS *N* TIMES DEPENDING ON *item\_name*." The Client also tests the value of the depends item to make sure it does not exceed the maximum specified in the OCCURS DEPENDING ON clause.

WARNING: Error in reading value of depends item 'name' in table 'name', using maximum of *nnnn* instead

This warning, which can occur during a `process` or a `clone` command, indicates that the Client detected a data error while retrieving the value of the depends item from the after-image of the record. A depends item is an item pointed to by an OCCURS DEPENDING ON clause, which is in the form "OCCURS *N* TIMES DEPENDING ON *item\_name*." The Client also tests the value of the depends item to make sure it does not exceed the maximum specified in the OCCURS DEPENDING ON clause.

WARNING: Filter for table 'name', which is not an OCCURS table, will be ignored

This message is an internal error that only occurs when there is an OCCURS table filter file "dbfilter.cfg" present in the config subdirectory for the data source. It indicates the filter references a table that is not an OCCURS table. The most likely cause of this error is that the filter and the client control table are not in sync. Try recreating the filter by either running the **makefilter** utility's `import` command or run a `redefine` command with the `-R` option (Redefine All) to try to remedy this situation. This situation should never occur, as whenever you run a `redefine` command or you run the Client Configurator **makefilter** is automatically launched. However, you need to pay attention to the cases where the program is unable to compile the filter due to errors in the filter's source file. Look in the **makefilter** log file to determine why the compile failed.

WARNING: `fixed_delay` scheduling overrides `daily` scheduling

See "`sched_delay_secs`" under [Scheduling] in the appendix the *Databridge Client Administrator's Guide*.

WARNING: Group name cannot be collapsed, as it contains an item with an OCCURS clause

A groups of unsigned NUMBER or ALPHA items can be collapsed into a single item whose length is the same as that of the GROUP. This operation works like a REDEFINE in GenFormat and it provides a simple way of dealing with dates that are represented as a DMSII GROUP. This operation is only allowed for GROUPs that do not contain items with OCCURS clauses. Attempting to collapse such a GROUP results in this WARNING.

WARNING: Group name cannot be collapsed; item name is not an {ALPHA item | unsigned NUMBER} [like the items that precede it]

A groups of unsigned NUMBER or ALPHA items can be collapsed into a single item whose length is the same as that of the GROUP. This operation works like a REDEFINE in GenFormat and it provide a simple way of dealing with dates that are represented as a DMSII GROUP. Attempting to collapse a GROUP whose items do not meet the above restrictions results in this WARNING.

WARNING: Illegal split offset for item 'name' in table 'name' will not be performed

This message is a result of a bad user script that attempt to split an item by specifying a bad offset. You will not see this error when using the Client Configurator as it will not allow you to specify an illegal offset.

WARNING: Index column 'name' in table 'name' has NULL attribute, this may cause updates to fail

This message indicates that the index includes a column that allows nulls. If you try to create a Primary Key constraint, it will fail. However, if you create an index, it will appear to work. Any update where this column is NULL, will fail; when the value is NULL, the where clause will be unable to satisfy the test for equality. In the case of a MISER database, the Client allows one column (which is a MISER date) and nulls to be included in an index. It does this by generating additional logic to test for NULL in the stored procedures that do updates and deletes.

WARNING: Index for tables in data set *name[/rectype]* contains a date that allows nulls; you will need to run a *redefine* command

This message, which is limited to MISER databases, indicates that the Client encountered an unregistered MISER date column that allows nulls. Running a *redefine* command will correct this problem, which only occurs when you upgrade to 6.2 in an environment where the NULL attributes of this column were set using user scripts. The configuration parameter *use\_nullable\_dates* enables this feature.

WARNING: Index 'name' for table 'name' has too many keys -- command terminated

This message, which can occur during a *define* or *generate* command, indicates that the client cannot use the selected DMSII SET as a source for an index, as the number of key would exceed the limit of the corresponding relational database. You need to either force the client to use AA Values (or RSNs) or create a composite key.

WARNING: Index name '*name*' already in use by another DataSource, using '*name1*' instead

This message, which occurs during a *define* or *redefine* command, indicates that an index name that was not unique across data sources was renamed. The new index names are constructed by appending the smallest possible integer (for example, 1, 2, 3,...) to the name to make it unique. If this causes the name to become too long, the last few characters of the index name are changed to digits.

WARNING: Index name '*name*' already in use, using '*name1*' instead

This message, which occurs during a *define* or *redefine* command, indicates that an index name that was not unique was renamed. For example, this message can occur when the index name gets truncated, therefore making it indistinguishable from another index name.

The new index names are constructed by appending the smallest possible integer (for example, 1, 2, 3,...) to the name to make it unique. If this causes the name to become too long, the last few characters of the index name are changed to digits.

WARNING: Insert for filtered secondary table '*name*' failed, changing it to a delete/insert (AFN=*afn*, ABSN=*absn*, SEG=*seg*, INX=*inx*)- Keys: *column\_name* = *value*, ...

This message is the standard error recovery indication from an exception that occurs during the processing of an insert for an OCCURS table that has a filter. It indicates that the data in the table is corrupted, this can occur during the fixup phase, at which time this message is not displayed unless the *-v* option is enabled.

WARNING: IO error writing to trace file "*trace\_filename*", file closed; *errno=number* (*errortext*)

This message can appear during any Client command when the *-t* option is enabled. It indicates that there was an I/O error writing to the specified trace file. The *errortext* which is generated by the system should explain why the error occurred.

WARNING: IPC connection to service lost, terminating command

This message, which can occur when using the service, indicates that the client has lost its IPC connection to the service and is therefore shutting down. This normally indicates that the service (or daemon) has crashed or was stopped.

WARNING: Item '*name*' in table '*name*' contains an illegal character in data [, record will be discarded - Keys: *columnname* = *value*...

This message, which is only applicable when using the translation DLL, indicates that the DLL detected an illegal character (such as an EBCDIC character that translates to a value that is reserved for the starting bytes of a DBCS character, which is encountered when not in double byte mode). Leaving this character in the data causes the next character to be interpreted as byte 2 of a DBCS sequence, which can lead to a SQL syntax error if this is the last character in the sequence (the closing quote will be lost). The DLL handles this error, and typically does so by changing such a character to a question mark (?). If the DLL determines that there are too many errors, it causes the record to be set to NULL, in which case the last clause of this message will be included.

WARNING: Item '*name*' in table '*name*' contains an illegal digit 0x*digit*, {number set to NULL value | record will be discarded} - Keys: *columnname* = *value*,...

This message can appear during a `process` or `clone` command. It indicates that the program encountered a value other than 0–9 (excluding the sign digit of a signed numbers). Numeric items that contain nothing but the digit 0xf are recognized as NULL; these are not flagged as in error except if they are keys or the parameter `discard_data_errors` is set to True, in which case the record is discarded.

WARNING: Item '*name*' in table '*name*' does not contain a valid GUID, value set to NULL

This message, which is limited to the SQL Server client, when storing an ALPHA(36) item as a uniqueidentifier data type, indicates that the data is not a valid GUID and will therefore be set to NULL to avoid causing a SQL error.

WARNING: Item '*name*' in table '*name*' has 8-bit characters in data[, { column set to NULL | record will be discarded} ] - Keys: *columnname* = *value*, ...

This message can appear during a `process` or `clone` command. It indicates that the Client has changed all 8-bit characters to question marks. The configuration parameter `inhibit_8_bit_data` determines whether 8-bit characters are stored in the data or flagged as errors. If the percentage of characters in error exceeds the value of the parameter `alpha_error_cutoff` the column is set to NULL instead. If the parameter `discard_data_errors` is set to true the record is discarded.

WARNING: Item '*name*' in table '*name*' has control characters in data[; too many errors column set to NULL] - Keys: *columnname* = *value*, ...

This message can appear during a `process` or `clone` command. It indicates that the Client has changed all control characters to question marks. The configuration parameter `inhibit_ctrl_chars` determines whether some or all control characters are stored in the data are flagged as errors. There are a few control character such NULL, TAB (when used as the bulk loader delimiter), CR and LF that cannot be included in the data. If you wish to preserve these characters you need to clone the column as a binary (BINARY for SQL Server and RAW in the case of Oracle). If the percentage of characters in error exceeds the value of the parameter `alpha_error_cutoff` the column is set to NULL instead.

WARNING: Item '*name*' in table '*name*' has control characters in data, record will be discarded - Keys: *keyname* = *value*, ...

This message can appear during a `process` or `clone` command, and it indicates that the Client encountered illegal control characters in the data. The configuration file `inhibit_ctrl_chars` parameter determines whether or not some control characters are stored in the database or flagged as errors. The Client discards the record because the `discard_data_errors` parameter is set to `True`.

WARNING: Item '`name`' in table '`name`' split into multiple columns

This message, which occurs during a `define` or `redefine` command, can occur if a DMSII ALPHA item is too big for the relational database to handle and the `DIOPT_Split_Item` bit (4) is set in the `di_options` column of the corresponding `DMS_ITEMS` table. When this happens, the item is stored in multiple columns. If the bit is not set, the item is truncated instead.

WARNING: Item '`name`' in table '`name`' was truncated to `ddd` characters

This message, which occurs during a `define` or `redefine` command, can appear if a DMSII ALPHA item is too big for the relational database to handle and the `DIOPT_Split_Item` bit (4) is not set in the `di_options` column of the corresponding `DMS_ITEMS` table. When this happens, ALPHA items that are too large are truncated.

WARNING: Key\_Item '`name`' in table '`name`' contains an illegal character in data, record will be discarded - Keys: `columnname = value,...`

This message indicates that the translation DLL detected illegal characters (such as an EBCDIC character that translates to a value that is reserved for the starting bytes of a DBCS character, which is encountered when not in double byte mode). Since this is item is a key, the Client tolerates no data errors. The record is written to the discard file and otherwise ignored.

WARNING: Key\_Item '`name`' in table '`name`' contains an illegal digit `0xdigit`, record will be discarded - Keys: `columnname = value,...`

The record is written to the discard file and is otherwise ignored. For more information on discard files, see "Special Handling of Key Items in Discard Files" in Chapter 3 of the *Databridge Client Administrator's Guide*.

WARNING: Key\_Item '`name`' in table '`name`' has 8-bit characters in alpha data, record will be discarded  
- Keys: `columnname = value,...`

The record is written to the discard file and otherwise ignored. For more information on discard files, see "Special Handling of Key Items in Discard Files" in Chapter 3 of the *Databridge Client Administrator's Guide*.

WARNING: Key\_Item '`name`' in table '`name`' has control characters in data, record will be discarded  
- Keys: `columnname = value,...`

The record is written to the discard file and otherwise ignored. For more information on discard files, see "Special Handling of Key Items in Discard Files" in Chapter 3 of the *Databridge Client Administrator's Guide*.

WARNING: Key\_Item '`name`' in table '`name`' is NULL, item will be treated as blank - Keys: `columnname = value, ...`

This message can appear during a `process` or `clone` command when the value of a key item of type ALPHA is NULL. If the `DSOPT_Keep_Null_Alpha_Keys` bit (128) of the `ds_options` column of the corresponding `DATASETS` entry is set, the program treats the NULL value as blanks instead of discarding the record. This warning is just a reminder that a NULL key was changed to blanks. This action can sometimes lead to duplicate records.

WARNING: Key\_Item 'name' in table 'name' is NULL, record will be discarded - Keys:  
columnname = value, ...

This message can appear during a `process` or `clone` command when the value of a key item is NULL. The program typically treats a NULL key as a non-fatal error; the record is written to the discard file and otherwise ignored. You are given the option to make the program treat NULL ALPHA keys as blanks instead of NULL. To do this, the `DSOPT_Keep_Null_Alpha_Keys` bit (128) of the `ds_options` column of the corresponding `DATASETS` entry must be set using user scripts.

WARNING: Link 'item\_name' values for DataSet *name[/rectype]* no longer valid

This message can occur during a `process` or `clone` command and indicates that the link items for the specified data set are invalid because the target data set's AA values were invalidated by a DMSII garbage collection or structural reorganization. You must reclone both data sets.

WARNING: Malformed AF\_header DOC record ignored

Databridge Engine and Enterprise Server use this type of DOC record to pass information about an audit file to the Client when they start processing updates from the given audit file. This message indicates that the record that has been sent is malformed. This is an internal error that should never occur. Contact Micro Focus Technical Support.

WARNING: Maximum discards threshold reached, aborting clone

This message indicates that the client has exceeded the limit on the maximum number of total discard records that was specified in the configuration file using the first value of the parameter `max_discards`. This has happened during data extraction and the clone will be aborted.

WARNING: Maximum discards threshold reached, {DBClient | dbutility} will stop at the next quiet point

This message indicates that the client has exceeded the limit on the maximum number of total discard records that was specified in the configuration file using the first value of the parameter `max_discards`. This has happened during the change tracking phase and the client will stop at the next quiet point.

WARNING: Maximum discard threshold reached for table 'name' additional discards will not be archived

This message indicates that the client has exceeded the limit on the maximum number of discard records for the given table that was specified in the configuration file using the second value of the parameter `max_discards`. This situation results in the client ignoring the record without writing to the discard file, as doing so could result in a huge discard file that would slow the client down considerably, as a result of continuously appending a record to the discard file and closing the file.

WARNING: Miser database requires "param name" to be set to value; parameter set to value

This message indicates that the specified configuration parameter (`miser_database=True`), which is required for a MISER database, has the wrong value. You must correct the value of the parameter `param_name` in the configuration by using the value specified in the message. These parameters include `automate_virtuals` (True), `flatten_all_occurs` (True), `use_nullable_dates` (True) and `default_date_format` (1)

WARNING: More than one nullable date found in index for table 'name', updates will fail if column name is NULL

This message occurs when using a MISER database with the SQL Server client and you have more than one MISER date being used as a key. The only solution is to use an RSN as the index instead of the SET.

WARNING: Multisource DataSet *name[/rectype]* must be recloned, run reorg command: *ds\_mode set to 34*

This message occurs during a *redefine* command when a multisource data source is reorganized. It indicates that the specified data source must be recloned. You may need to wait until other data sources are caught up with the reorg before doing this. However, the *reorg* command can determine if the data source can be recloned.

WARNING: Multisource DataSet *name[/rectype]* processing stopped until other sources are reorged - mode 35

This message, which can occur during a *process* or *clone* command involving a multisourced data sets, indicates that the data set has a mode of 35. This means that the source is waiting for all other data sources to be reorganized.

WARNING: Multi-threaded updates need at least 2 DMSII buffers per thread to be effective

This message, which only occurs when processing a text-based configuration file, indicates that the number of DMSII buffers must be increased to support multi-threaded updates. For best results, set the parameter *n\_dms\_buffers* to 0 to let the Client determine the appropriate value.

WARNING: Name '*name*' is an SQL reserved word, using '*newname*' instead

This message can occur during a *define* or *redefine* command. It indicates that a table, column, or index name is a SQL reserved word (for example, *order* or *update*) and that the specified new name is being used in its place. The *newname* is the original name with a suffix of *\_x*.

If using the SQL Server Client and the object in question is a table, you can force the Client to use this name by setting the configuration parameter *bracket\_tabnames* to true.

WARNING: No data received from {DBServer | DBEnterprise Server} for *number* minutes

If the Client waits for TCP/IP input from the Databridge server past a certain threshold, the watchdog timer thread returns this message. The initial threshold is 5 minutes. This message is then reissued (after 15 minutes and then 30 minutes of inactivity and every 30 minutes thereon) until the Client receives input from the Databridge server. The timer is reset when the Client receives input. When you get this message make sure to check whether the Databridge software requires an operator response on the mainframe.

WARNING: No DataSources found in client control tables

This message which can occur during an *unload* command, when a data source name of *\_all* is specified. It indicates that the *dbutility* program did not find any data sources to unload.

WARNING: No items except keys were active in occurs table '*name*', active column set to 0

This message can occur during a *generate* command. It indicates that the Client disabled cloning of the specified secondary table because it contained nothing but keys.

WARNING: No items were active in table '*name*', active column set to 0



This message can occur during a `generate` command. It indicates that there are no data items to be cloned for the specified table. This message can occur in the following circumstances:

When you set all of the active columns in the `DATAITEMS` table to 0. Unless you intended for all the active columns to be set to 0, reset the active columns and run the `process` again.

When you change a table name in the `DATATABLES` Client control table but fail to make the same change in the `DATAITEMS` table.

WARNING: No updates received from {DBServer | DBEnterprise Server} for number minutes

If the Client is using `DBWait` and it uses a non-zero value for the second parameter of `max_wait_secs`, you end up with frequent data on the line which acts pretty much like a heart-beat. However, this makes the "No data received from {DBServer | DBEnterprise Server} for number minutes" message disappear. We have added this message so that the watchdog timer can issue warnings about the lack of updates, which could be indicative of a problem on the MCP side of things. The initial threshold is 5 minutes. This message is then reissued (after 15 minutes and then 30 minutes of inactivity and every 30 minutes thereafter) until the Client receives an update from the Databridge server. The timer is reset when the Client receives an update. When you get this message make sure to check whether the Databridge software requires an operator response on the mainframe.

WARNING: No value specified for "max\_temp\_storage" in config file, the default value will be used

(Windows only) This message appears only when you explicitly set `max_temp_storage` to 0. To correct this error, change the `max_temp_storage` back to the default setting.

WARNING: Open failed for script file "name", errno=number (errortext)

This message can occur during a `process` or `clone` command when the `check_user_scripts` parameter is enabled.

WARNING: Overflow error in translation DLL for Item 'name' in table 'name'[, record will be discarded] - Keys: `columnname = value, ...`

This message can appear during a `process` or `clone` command. It indicates that the character translation DLL called by the Client returned an overflow error status. If the configuration parameter `discard_data_errors` is set to `True`, the Client discards the record.

WARNING: Overflow error in translation DLL for Key\_Item 'name' in table 'name', record will be discarded - Keys: `columnname = value, ...`

This message, which can appear during a `process` or `clone` command, indicates that the character translation DLL called by Client returned an overflow error status while translating the data for an item that is a key. This is not a fatal error; the record is written to the discard file and otherwise ignored. See "Special Handling of Key Items in Discard Files" in Chapter 3 of the *Databridge Client Administrator's Guide*.

WARNING: Parameter "max\_retry\_secs" must not exceed the 2'nd value of "max\_wait\_secs"; parameter set to nn

When you specify a second value for the parameter `max_wait_secs`, this value gets used in the request sent to the Engine. The value of the parameter `max_retry_secs` needs to be smaller than this value for this to make sense. If the value of `max_retry_secs` is greater than the second value of the parameter `max_wait_secs` the client forces the value of `max_retry_secs` to be ½ the value of the second part of `max_wait_secs`.

WARNING: Parameter `use_ctrl_tab_sp` requires that `aux_stmts` be greater than 0 - `use_ctrl_tab_sp` parameter disabled.

This warning indicates that the parameter `aux_stmts` is zero; the parameter `use_ctrl_tab_sp` cannot be set to True if there are no database API stmt structures required to support the host variables. When the parameter `use_ctrl_tab_sp` is True, (default) all of the control tables updates use host variables. We recommend that you don't change the value of the deprecated parameter `use_ctrl_tab_sp`.

WARNING: Parameter `use_ctrl_tab_sp` requires that `aux_stmts` be greater than 0 - `use_ctrl_tab_sp` parameter disabled.

This warning indicates that the parameter `aux_stmts` is zero; the parameter `use_ctrl_tab_sp` cannot be set to True if there are no database API stmt structures required to support the host variables. When the parameter `use_ctrl_tab_sp` is True, (default) all of the control tables updates use host variables. We recommend that you don't change the value of the deprecated parameter `use_ctrl_tab_sp`.

WARNING: Parameter "use\_global\_stateinfo" requires that "optimize\_stateinfo" be enabled - "optimize\_stateinfo" parameter enabled.

This message appears during any Client command if you disable the deprecated parameter `optimize_stateinfo` without disabling the deprecated parameter `use_global_stateinfo`. For more information, see `preserve_deletes` in the appendix of the *Databridge Client Administrator's Guide*. For optimal efficiency of the Client, avoid changing the values of deprecated parameters.

WARNING: Parameters "convert\_ctrl\_char" and "inhibit\_ctrl\_chars" are mutually exclusive

This message can occur when a text configuration file is being processed. It indicates that you need to set one of the specified parameters to False.

WARNING: Parameters `use_decimal_aa` and `use_binary_aa` are mutually exclusive

This message can occur when a text-based configuration file is being processed. It indicates that the specified parameters are mutually exclusive and that you need to set one of them to False.

WARNING: Parent\_AA values for DataSet `name[/rectype]` no longer valid

This message can occur during a `process` or `clone` command and indicates that the Parent\_AA values for the specified embedded data set are no longer valid because the DMSII database was reorganized. You must reclone both data sets.

WARNING: Pending DMSII REORG -- client `update_level = nnn`

This message, which applies to a `process` command indicates that the client's update level is not the same as the DMSII update level, which is shown in the DB\_Info lines in the log file at the start of the run. The client will soon encounter a DMSII reorg as it processes audit files.

WARNING: Performance statistics cannot be disabled when multi-threaded updates are being used, parameter set to true

When running multi-threaded updates, you must have performance statistics enabled, as the Client uses the table statistics to balance the load between threads. Except for `show_table_stats`, statistics parameters should be left enabled.

WARNING: 'quit' command is only meaningful during `process` or `clone` commands

This message indicates that the operator issued a `dbutility QUIT` command. The program only honors this command during a `process` or `clone` command.

WARNING: 'quit' command will take effect at the start of audit file processing

This message indicates that the operator issued a `dbutility QUIT` command during data extraction. The `QUIT` command is not acted on during the data extraction phase; it will be acted on at the start of audit file processing.

WARNING: Received unexpected result: `seq_no = hexnumber`; expected `seq_no = hexnumber`

This message indicates that the host sent an unexpected response to the Databridge Client. Typically, this occurs when the previously executed Databridge command had an error. Since this happens often at session startup where the expected sequence number is 1, this message is not printed when the sequence number is 1 unless the `-v` option is enabled. This message is displayed unconditionally for any other expected sequence number as this is a symptom of a protocol error between the Databridge server and the Client.

The Databridge Client typically recovers from this condition by discarding the unexpected message, which is left over from a previous session.

WARNING: `Redefine` command found nothing to do

The `redefine` command is driven by the bit settings of `status_bits` and `active` column in the `DATASETS` Client control table. This message indicates that the command found nothing in the `status_bits` columns that indicated that something needed to be redefined or remapped and that all the data sets that needed mapping had their `active` column set to 0. If you want the command to do something, except for the processing of changes caused by DMSII reorganizations, you need to set the `DS_Needs_Remapping` bit in the `status_bits` column of the data sets that you want remapped. In the case of data sets that were not previously mapped, setting the `active` column to 1 prior to running the `redefine` command is all that is needed. Alternatively, you can add the `-R` command line switch to force all data sets to be redefined. If you execute the `redefine` command without doing these things, this message appears to remind you that nothing happened.

WARNING: `Reorg` command found nothing to do -- there are no DataSets with mode 31

This message indicates that you did not have to run the `reorg` command, as no data sets whose mode is 31 were found. It is mainly meant to provide some feedback when the command is executed from the Client Console.

WARNING: `Session initialization script name` failed

This message indicates that the specified script failed. Check the preceding API or SQL error to see why it failed.

WARNING: `Set name` is not usable as an index as it has too many keys

This warning occurs if you have a DMSII SET that has more keys than the relational database allows in an index. Consider using an RSN. The limit on the number of keys in an index is 16 keys for SQL Server and 32 keys for Oracle.

WARNING: `Set name` of data set `name[/rectype]` allows key changes

This warning appears when a data set whose optimum SET has the `KEYCHANGEOK` attributes is encountered (during a `define` or `redefine` command) when using Databridge Engine 6.0 or earlier. It is intended to warn you of the resulting increase in TCP/IP overhead. With the older Engines, the Client requests that all `MODIFY` records be passed to the Client as `MODIFY BI/AI` pairs. With the newer Engines, this decision is made by the Engine after comparing the old and new values of the keys in the index. The Client must register the keys it uses with the Engine in order for the Engine to perform this test.

WARNING: Setting `n_update_threads` to 1 will not improve performance, set it to 2 or higher

Using one update thread makes the program work in single-threaded mode, except that all updates are queued to the worker. There is almost no overlap in SQL execution, as the main thread only handles the updates of the DATASETS table. The program will queue updates to the update worker and use all available DMSII buffers. Using one update thread is not likely to help performance.

WARNING: SQL operation [for table '*name*'] appears to be hung (elapsed time *eee*, incident # *nn*)

This message is returned by the watchdog timer thread when the client wait time for a SQL operation to complete reaches the primary threshold specified by the parameter `sql_exec_timeout`. The message is issued at increasing intervals after this point until it reaches 30 minutes, at which point it repeats every 30 minutes. This can indicate that a badly written database application program has locked the table. If this is case, determine the program and terminate it. This problem can also be caused by issuing a select statement that has no "where" clause for a very large table.

If the table name is known to the client, it is included in the message. The value *eee* is expressed in the appropriate units based on its value (e.g., 5 minutes or 15 seconds). The value *nn* is a thread specific counter that is used to make it easy to see if this is a new problem or a repeat warning for an existing hang.

WARNING: Stored procedures required for data set *name*[/*rectype*] which has a key that allows nulls; `DSOPT_No_StoredProcs` option will be ignored

This message, which only applies to MISER databases, indicates that you are attempting to disable the use of stored procedures for a data set that has a key that is a MISER date. This is not possible, as the stored procedures contain logic to handle the update when the key is NULL. If you set the configuration parameter `use_stored_procs` to False and run a `redefine` command (with the /R option) you will see several occurrences of this message. The client will then ask you to run a `reorg` command, which will fix things up and reset the above mentioned bit for such data sets; after that you will not see this message again.

WARNING: Table '*name*' cannot be altered data truncation would occur

This message indicates that the size of a column has decreased as a result of a reorganization or a change in a user script. The relational database will not let you do an `alter` command that does something like changing a column from a NUMBER(10) to a NUMBER(8).

WARNING: Table '*name*' cannot be altered, this type of alter is not supported

This message indicates that it is not possible use an `alter` command to fix a column whose data type has changed. You may be able to handle this manually by creating a new columns and populating the new column using the old values. The only other alternative is to reclone.

WARNING: Table '*name*' contains a column of type CLOB which requires host variables; active set to 0 for DataSet *name*[/*rectype*]

(Oracle only) This message indicates that you're updating a column whose data type requires the use of host variables. This message occurs if the client encounters a table that contains a column of type CLOB, while `aux_stmts` is set to 0 or the table has the `dt_options` bit `DOPT_No_Aux_Stmts` set. You'll need to correct this before going any further.

WARNING: Table '*name*' contains a column of type CLOB which requires host variables; `minimize_col_updates` option disabled for DataSet *name*[/*rectype*]

(Oracle only) This indicates that the client is updating a column whose data type requires the use of host variables. The client will issue this warning if you set the configuration parameter `optimize_col_updates` to True or the corresponding bit in `ds_options` of the DATASETS table entry is set for a column of type CLOB. In response, the Client will disable the use of host variables for this table and process the updates for this table if the aforementioned bit is zero.

WARNING: Table '*name*' contains more than one nullable key that is a MISER date - updates will fail if the column *col\_name* is null

This message, which is only applicable to MISER databases, indicates that you have more than one MISER date that allows NULLS in the index. The Client only handles one such key. To get past this error, reset the `da_options` bit `DAOPT_Nulls_Allowed` (1). However, if the date in question happens to be NULL when you attempt to update the table, the update will fail. Then, the solution is to use an RSN or a composite key as the index.

WARNING: Table name '*name*' already in use by another DataSource using '*newname*' instead.

This message can occur during a `define` or `redefine` command. It indicates that specified table name was already in use by another data source. The new table name is constructed by appending the smallest possible integer (for example, 1, 2, 3,...) to the name to make it unique. If this causes the name to become too long, the last few characters of the table name are changed to digits.

WARNING: Table name '*name*' already in use, using '*newname*' instead.

This message can occur during a `define` or `redefine` command. It indicates that a table name that was not unique was renamed. This is different from tables being renamed when they are the result of a split data set. For example this message can occur when the table name gets truncated, therefore making it indistinguishable from another table name.

The new table names are constructed by appending the smallest possible integer (for example, 1, 2, 3,...) to the name to make it unique. If this causes the name to become too long, the last few characters of the table name are changed to digits.

WARNING: Table '*name*' exceeds max record size of *ddd* bytes; table '*name1*' will be used for additional columns

This message applies to the Databridge Client for Microsoft SQL Server, and it can occur during a `define` or `redefine` command. It indicates that the specified table was split because the maximum size limit for a relational database record was reached. Keys in the primary table are replicated into the split table; the tables are not linked in any way. See "Relational Database Split Tables" in Chapter 5 of the *Databridge Client Administrator's Guide*.

WARNING: Table '*name*' has more than *ddd* columns; table '*name1*' will be used for additional columns

This message can occur during a `define` or `redefine` command. It indicates that the specified table had to be split because the maximum column limit (*ddd*) for the relational database was reached. The maximum number of columns in a table is 1024 for Microsoft SQL Server and 1000 for Oracle. The keys in the primary table are replicated into the split table; the tables are not linked in any way. See "Relational Database Split Tables" in Chapter 5 of the *Databridge Client Administrator's Guide*.

WARNING: Table '*name*' has no index the DataSet *name[/rectype]* cannot be tracked, mode set to 11

This message is returned when a table for a DMSII data source has no index, in the past such data sets had their `ds_mode` set to 11 with no indication that this happened. This message rectifies this situation. You need to create a composite key to be able to track changes to this data set.

WARNING: Table '*name*' is a split table; this type of reorg is not supported

The Databridge client issues this message when a split table is reorganized. This type of reorganization is not supported when a column are added or deleted to the split table. You must reclone the data set.

WARNING: Table '*name*' is using a PRIMARY KEY, index column '*name*', which is a MISER date, cannot have the NULL attribute

This message, which is only applicable to Miser database users, indicates that you are attempting to use a Primary Key constraint as the index for a table that has keys that are MISER date columns and allow nulls. To make the client use a unique index instead, you must clear the `dt_options` bit `DTOPT_PrimaryKey (0x800)`.

WARNING: `Tcptest` length exceeds maximum value of *maxlen*, using *maxlen* instead

This warning indicates that the `tcptest` length parameter specified on the command line is too large. The value of *maxlen* (currently 128K) is used instead.

WARNING: The clone of DataSet *name*[/*rectype*] will be deferred until the client catches up with all pending DMSII reorgs

This warning, which appears at selection time during a `process` or `clone` command, indicates that the Client has not yet caught up with outstanding DMSII reorganizations and the requested clone cannot be performed at this time. The clone will be performed as soon after the Client processes that last outstanding DMSII reorganization.

WARNING: The configured number of `stmts` (*nn*) is insufficient for *k* threads

When using multi-threaded updates, you need a minimum of 20 database API STMTs plus the number of threads. When not using multi-threaded updates, the minimum number of API STMTs required is 20.

WARNING: The filtering status of data set *name*[/*rectype*] has changed from *n* to *m*; `ds_mode` set to 0 to force it to be recloned

This message indicates that an OCCURS table that was not previously filtered is now filtered (0 > 1) or vice versa. Since this will change the number of records stored in the table recloning is the only option.

WARNING: The "`minimize_col_updates`" parameter will be set to false, as it is invalid for a MISER database

When the Client loads the control tables, if the `miser_database` parameter is set to True, the Client disables the `minimize_col_updates` parameter (if set to True) and issues this warning.

WARNING: The parameter '*name*' is deprecated; it will be ignored

The Client issues this message if it finds one of the following deprecated parameters in a text configuration file: `auto_reclone`, `bcp_date_format`, `bcp_delim`, `check_key_changes`, `encode_passwords`, and `redef_all_active_ds_n_bcp_threads`. The Oracle client will also issue this warning if you attempt to use the `bcp_delim` parameter. Also using the old parameters `optimize_col_updates` and `enable_optimized_col` will generate a similar message stating the new name of the parameter in question.

WARNING: This command will drop all Databridge tables, do you really want to do this?- To bypass this check, use the 'u' option for the dropall command

This message appears during a `dropall` command when you use the command without the `-u` option. The `-u` option is required as a safeguard against accidentally dropping the Databridge tables.

WARNING: Too many data errors for DataSet *name[/rectype]* -- additional errors will not be {displayed | logged}

This message can appear during a `process` or `clone` command. It indicates that the count of data errors for the given data set has just crossed the display or log cutoff threshold and that no more data errors will be displayed or logged for this data set. See `error_display_limits`.

WARNING: "trace 1" command ignored as this would create a second copy of the log file

This message appears when a `dbutility` console TRACE command specifies a trace mask of 1 or "verbose" is enabled. The Client won't process this option unless additional bits are set in the trace mask, resulting in a trace file that can be differentiated from the log file.

WARNING: Transaction group aborted; *number* SQL operations rolled back

This message, which is only written to the trace file when the trace file TR\_VERBOSE bit in the trace mask is set, occurs when the Client aborts some updates at the end of the last audit file processed. These updates are picked up when the next audit file is processed.

WARNING: Truncation error in translation DLL for [Key\_] Item '*name*' in table '*name*'[, record will be discarded] - Keys: *columnname* = *value*,...

This message can appear during a `process` or `clone` command. It indicates that the character translation DLL called by Client returned a truncation error status. This is not a fatal error. However, when the configuration parameter `discard_data_errors` is set to True or the item is a key, the Client discards the record. If the item is a key, the Client also discards the record.

WARNING: Unable to clear 'changes' column in {DATASETS | DMS\_ITEMS} for data source *name*

This warning can occur when running a `redefine` command or when using the Client Configurator. It indicates that the Client was unable to clear the changes column for the entries in the specified table. Check the SQL error that precedes this warning to determine why the update failed. This column is used by the Client Configurator to highlight the items that have changed.

WARNING: Unexpected held buffer for DataSet *name[/rectype]*; buffer discarded

This warning indicates that an internal error occurred causing a before image record to not be immediately followed by an after image record. Contact Micro Focus Technical Support if you ever get this error.

WARNING: Unknown translation error (0xdd) for [Key\_]Item 'name' in table 'name'[, record will be discarded] - Keys: *columnname = value, ...*

This message can appear during a `process` or `clone` command. It indicates that the character translation DLL called by the Client returned an undefined error status of 0xdd. This message is a catchall for any new error status that may be added to the data translation DLL. This is not a fatal error. However, when the configuration parameter `discard_data_errors` is set to True or the item is a key, the Client discards the record.

WARNING: Update for filtered secondary table 'name' failed, changing it to a delete/insert (AFN=*afn*, ABSN=*absn*, SEG=*seg*, INX=*inx*) - Keys: *column\_name = value, ...*

This message is the standard error recovery indication from an exception that occurs during the processing of an update for an OCCURS table that has a filter. It indicates that the data in the table is corrupted, this can occur during the fixup phase, at which time this message is not displayed unless the `-v` option is enabled.

WARNING: Update for table 'name' had no matching row, changing it to a delete/insert (AFN=*afn*, ABSN=*absn*, SEG=*seg*, INX=*inx*) - Keys: *columnname = value, ...*

This message can appear during a `process` or `clone` command. It indicates that the executed update SQL statement for a user table resulted in a row count of zero, indicating that the update did not find any matching rows. Furthermore, the update involved an item with an OCCURS clause, so instead of just turning the update into an insert, the program first deletes any partial data that may be in the occurs table and inserts the after image of the update.

WARNING: Update for table 'name' had no matching row, changing it to an insert (AFN=*afn*, ABSN=*absn*, SEG=*seg*, INX=*inx*) - Keys: *columnname = value, ...*

This message can appear during a `process` or `clone` command. It indicates that the execution of an update SQL statement for a user table resulted in a row count of zero, indicating that the update did not find any matching rows. The program turns these updates into inserts and tries them again.

WARNING: User index creation script "*name*" failed

This message indicates that the execution of the index creation user script for the specified table (`script.user_index.tablename`) failed. See the preceding database API error message to determine the cause of the failure. The most likely reason is that the script is in error.

WARNING: User script directory must reside in the global working directory or the data source's working directory

When security is enabled all user scripts must reside inside the Client working directory. You will get this warning if you set the `user_script_dir` configuration parameter to point to a directory that does not meet this restriction.

WARNING: User table cleanup script *name* failed

This message can occur during a `process` or `clone` command and indicates that the specified script failed. Cleanup scripts are used to remove records from special tables that cannot be dropped.

WARNING: Value of item 'name' in table 'name' exceeds maximum allowed, value set to NULL



This message indicates that the value of a DMS item of type REAL (n) or REAL (m,n) is too large for the client to be able to handle. The client uses the precision and scale to create the corresponding relational database column, therefore attempting to use such a value could result in an SQL error.

WARNING: Value *mmm* for "maximum\_columns" too large for {Oracle| SQL Server} *Vm,Vn* using *nnn* instead

When the Client determines which version of SQL Server or Oracle it is connected to, it checks the value of the configuration parameter `maximum_columns` to make sure that this value does not exceed the maximum number of column supported by the database. You will only get this error if you use a very old version of SQL Server or Oracle.

WARNING: Variable Format DataSet *name[/rectype]* has links -- it will not be tracked

The Databridge client cannot track changes to variable format data sets that contain links. If the configuration parameter `track_vfds_nolinks` is set to False, the Client will not track any updates to these data sets. This message is a reminder about the action that the Client is taking.

WARNING: Variable Format DataSet *name[/rectype]* has links -- link updates will not be tracked

The Databridge Client cannot track changes to variable format data sets that contain links. If the configuration parameter `track_vfds_nolinks` is set to True, the Client will track non-link updates to these data sets. This message is a reminder about the action that the Client is taking.

WARNING: Virtual DataSet *name* has no index; all DataSets involved with it will not be selected

This message can occur during a `process` or `clone` command and indicates that no data sets associated with the given virtual data will be selected, because the virtual data set has no index.

WARNING: Virtual DataSet *name* will not be selected because its parent structure was not selected

This message can occur during a `process` or `clone` command and indicates that the virtual data set will not be selected because the data set from which it was derived was not selected.

WARNING: You are about to overwrite your input configuration file "*name*"  
- To bypass this check, use the 'u' option for the {import\_ | export} command

This message indicates that the input and output filenames are the same for an `import` or `export` command. If you want to convert a text configuration file named `dbridge.cfg` to a binary file, you must set the `-u` option in addition to specifying the input file using the `-f` option.

WARNING: You are attempting to run a `createscripts` command for a data source that was not created using the Client Configurator - The control tables need to be upgraded by running "`dbscriptfixup`" before you can safely do this

This message, which only occurs when you run a `createscripts` command, indicates that the client control tables are not compatible with the Client Configurator. You must first run the `dbscriptfixup` program; this will fix the client control tables and create new set of user scripts that reflect changes to be implemented. The Client Configurator uses an additional bit in the various `xx_options` columns in the control tables and the `xx_bmask` columns to keep track of table entries that were changed during customization.

WARNING: You are attempting to run a `redefine` command with the parameter "`use_dbconfig`" set to true - The control tables need to be upgraded by running "`dbscriptfixup`" before you can safely do this

This warning is a reminder to run the `dbscriptfixup` program before using the Client Configurator. The `dbscriptfixup` program fixes the control tables and creates a new set of user scripts from which you can recover a corrupted relational database. It also sets the `use_dbconfig` parameter in the binary configuration file to True. This parameter is required for the Client Configurator to run. (Attempting to edit this or other parameters in the binary configuration file will only corrupt the file.)

## Messages Associated with Client Exit Codes

Exit codes correspond to expected conditions (that is, they are not errors). These messages should be referred to instead of the host messages that often overlap them.

Exit code messages appears at the end of every client run and between restarted `dbutility` process commands as a result of scheduling or the built-in error recovery loop. This text (`exit_code_text`) appears in the form "`Client exit code: dddd(nnn) - exit_code_text`".

For a list of Client Exit Codes, see Appendix E of the *Databridge Client Administrator's Guide*.

- 0000 - Successful
- 0001 - General error
- 0002 - Database reorganized
- 0094 - DBServer STOP TIME limit
- 0093 - DBServer STOP TASK encountered
- 1015 - Operator terminated the worker
- 1135 - Operator terminated DBEnterprise
- 2001 - Command line usage error
- 2002 - Client control table error
- 2003 - Client control table locked
- 2004 - Client control table load error
- 2005 - Data source not found error
- 2006 - Redefine command required
- 2007 - Could not connect to DBServer
- 2008 - Invalid data set name
- 2009 - Invalid `ds_mode` value
- 2010 - Create/cleanup error
- 2011 - Error dropping index
- 2012 - Bad structure index received from DBServer
- 2013 - Memory allocation error
- 2014 - No active structures
- 2015 - No remaining active structures
- 2016 - Load error
- 2017 - Load or index creation error
- 2018 - Index creation error
- 2019 - Fixup phase deferred
- 2020 - Client operations inhibited
- 2021 - Operator issued quit
- 2022 - SQL error
- 2023 - COMMIT TRANSACTION error
- 2024 - ROLLBACK TRANSACTION error
- 2025 - AFN limit reached
- 2026 - EBCDIC-ASCII Library error
- 2027 - Improperly linked virtual dataset
- 2028 - Not all virtual dataset sources specified
- 2029 - Records have been discarded, see log for details

2030 - Database signon error  
2031 - Verification count discrepancy  
2032 - Generate command required  
2033 - Reorg command required  
2034 - DMSII reorg caused some AA values to become invalid, add the -y option to the process command to force a reclone  
2035 - Aborted clone  
2036 - Operator issued abort  
2037 - Relational database deadlock  
2038 - End of fixup phase for cloned DataSets  
2039 - Global working directory not found  
2040 - Lock file open error  
2041 - Install directory not found  
2042 - Redefine command required  
2043 - IO error writing script files  
2044 - DMSII link improperly setup in control tables  
2045 - Reorg script error  
2046 - Refresh of stored procedures failed  
2047 - Bulk Loader failure  
2048 - Missing binary configuration file  
2049 - Bad checksum in configuration file  
2050 - IO Error reading configuration file  
2051 - Configuration file processing error  
2053 - Error in writing discard file  
2054 - Discard threshold exceeded  
2055 - 2056 (Not currently used)  
2057 - Wrong value for parameter bcp\_decimal\_char  
2058 - SQL operation cancelled  
2059 - Server connection timed out  
2060 - Commit CHECKPOINT FREQUENCY parameters are all 0  
2061 - Error in loading DLL or finding entry points  
2062 - Error in updating control table  
2063 - Error creating control table  
2064 - Error dropping control table  
2065 - Malformed unload file  
2066 - Error dropping user table  
2067 - Control tables are incompatible with DBConfig, you need to run dbscriptfixup  
2068 - Unable to create directory  
2069 - Unable to allocate STMT; try reducing the value of aux\_stmts  
2070 - Error creating file  
2071 - User script in error  
2072 - Bad DMSII database timestamp  
2073 - History table error  
2074 - Data source already defined  
2075 - Index for user table has too many columns  
2076 - Mismatched AFNs in control tables  
2077 - Protocol error  
2078 - File does not exist  
2079 - IO error reading filter file  
2080 - Malformed binary filter file  
2081 - Bad checksum in binary filter file  
2082 - Syntax error in filter source file  
2083 - Filter generation failed  
2084 - Unsupported table encountered in filter source file

2085 - Data source already exists in relational database  
2086 - 2090 (Not currently used)  
2089 - Database connection lost  
2091 - Databridge service failed  
2092 - Connection to Server closed by peer  
2093 - Connection to Server reset by peer  
2094 - Transport error for connection to server  
2095 - RPC timeout  
2096 - Initialization error  
2097 - Error processing RPC data  
2098 - Communications or protocol error  
2099 - Premature termination of RPC  
nnnn - Abnormal termination - see log for details. This message addresses situations other than a host error or any of the preceding exit codes.

# 3 Databridge Client Manager Service/ Daemon Messages

- ♦ “Log Messages” on page 189
- ♦ “Error Messages” on page 195
- ♦ “Windows Application Log Messages” on page 203
- ♦ “Client Console and Configurator Messages” on page 205

## Log Messages

The following messages are only written to the Client Manager Service log file, which is located in the `logs` directory in the global working directory. These messages are prefixed by the timestamp `hh:mm:ss` (where `hh` is hours, `mm` is minutes, and `ss` is seconds).

```
Bindsocket({AF_INET | AF_INET6}, portnumber)
```

This message, which is logged at the start of the service, indicates that the service is binding the socket to listen for connect requests on the specified port number using the given protocol. `AF_INET6` is IPV6 and `AF_INET` is IPV4.

```
Closing [{CNS | IPC}] Session dd, socket = nnn[, userid = 'userid ']
```

This message indicates that the corresponding Console or IPC socket is being closed. IPC sockets are used to pass messages between spawned Client runs (DBClient or DBCntCfgServer) and the service/daemon (DBCntControl). The `userid` is provided for Console Sessions.

```
CNS Session dd, socket = nnn initialization complete
```

This message indicates that a console session has successfully been initialized. After the signon RPC is received and validated, the following message appears ("`CNS Session nn, ... signon complete`").

```
CNS Session nn, userid = 'userid', role = role, signon complete
```

This message indicates that a Console session has successfully signed on the service. The supplied `userid` and `role` are listed in the log file for reference. The defined roles are “administrator”, “operator” and “user”.

```
CNS Session nn, userid = 'userid', security = 0xhhhhhhh,0xhhhhhhh, signon complete
```

This message indicates that a Console session has successfully signed on the service. Instead of using a predefined role the `userid` is using customized security with the given attributes, represented as two 32-bit numbers displayed in hexadecimal format. Refer to Client Console documentation for a definition of these bits.

```
Configuration information read from binary file "dbcontrol.cfg"
```

This message, which is logged when the configuration parameter `debug` is set to `True`, confirms that the service has successfully read the binary configuration file `dbcontrol.cfg` from the `config` subdirectory.

```
CreateProcess: cd = "workingdir"
```

This message is logged when the configuration parameter `debug` is set to `True`. It shows the working directory used by a spawned run.

```
CreateProcess: cmdline = "cmdline"
```

This message is logged when the configuration parameter `debug` is set to `True`. It shows the command line for a spawned run.

```
CreateProcess: module = "filename"
```

This message is logged when the configuration parameter `debug` is set to `True`. It shows the program file used to spawn a run.

```
Current activity: nnn console sessions, mmm ipc sessions
```

This message shows the number of active sessions for Consoles and spawned runs. Each spawned run uses one IPC connection.

```
Data Source name disabled (client exit_code = nnnn)
```

This message indicates that the service disabled the specified data source because the run's exit code indicates an unrecoverable error. The reason for disabling a data source is to prevent any further Client runs in situations where the client will keep encountering the same problem. After the problem is resolved, you must enable the data source from the Client Console and restart the Client.

```
Data Source name disabled after number retries failed
```

The service automatically performs error recover for most exit codes that are thought to be recoverable. It does so by restarting the client after a brief delay and repeats this process a few times. If the client keeps failing it gives up after the `max_retries` value for the data source is reached. When this happens it disables the data source and logs this message.

```
Databridge Client Control Program Version version
```

This message identifies the version of the code. It is logged when the program starts.

```
DBCIntControl automatically launching a {generate | redefine} command for data source name
```

This message is logged when the `auto_redefine` or `auto_generate` options are enabled for a data source. It indicates that the service has launched a `DBCIntCfgServer` run to perform the specified command. These options are designed to automate the handling of simple DMSII reorganizations without interrupting production.

```
DBCIntControl automatically launching dbfixup for data source name
```

This message indicates that the service encountered a client exit code of 2056 that indicates that the Client Control Tables are out of date and need to be updated by running the `dbfixup` utility. Starting with version 6.6 the Client Manager Service automatically does this.

```
DBCIntControl shutting down; listen socket closed
```

This message is logged when the service shuts down and its associated socket closes.

```
Deleting working directory tree for data source name
```

This message is logged when a request is received from the Console to remove a data source and its directories.

```
End_of_run: data_source = name, process_id = nnnn, exit_code = dddd [(Run terminated prematurely)]
```

This message is logged when a launched Client notifies the service that the run is terminating. To determine which type of run it is, you must find the corresponding "Start\_of\_run: ..." message, with a matching process\_id. In the unlikely case when the Client run crashes or it loses the IPC connection to the service, the exit code will be followed by the text "Run terminated prematurely".

```
IPC Session dd, socket = nnn initialization complete
```

This message indicates that a launched run has initialized its IPC session and the two programs can now communicate with each other using this socket.

```
Launching script: "filename"
```

This message indicates that the specified command file (shell script for UNIX) is being launched by the service.

```
Manager for data_source name timed out -- stopping DBClntCfgServer
```

When the Console connects to the service, it requests information about the data source that the service does not have. The service launches DBClntCfgServer to run a manage command, which allows the service to indirectly get access to the database. To avoid tying up resources, such runs are brief in duration and terminate after a minute of inactivity. This message indicates that the launched run is terminating.

```
Non-standard data source working directory for srcname, files not deleted
```

This message is logged when processing a request from the Console to remove a data source's working directory. If it is not a subdirectory of the service's working directory, it is not removed and this message is logged. This prevents the deletion of a directory that has been moved to a different location for safekeeping purposes. We recommend that you always use the directory structure imposed by the service.

```
OPER: cmd command for data source src issued from console, userid = 'name'
```

This message indicates that the operator, whose userid is name, has issued the given Client console command for the given data source. This allows the auditing of operator actions to include actions that are directed at launched Clients.

```
OPER: Data Source name {enabled | disabled} by the console, userid = 'name'
```

This message indicates that the operator, whose userid is name, has enabled or disabled the given data source.

```
OPER: Data Source src added from console, userid = 'name'
```

This message indicates that the operator, whose userid is name, has added the specified data source.

```
OPER: Data Source src removed from console, userid = 'name'
```

This message indicates that the operator, whose userid is name, has removed the specified data source.

```
OPER: Data Source src signon information modified from console, userid = 'name'
```

This message indicates that the operator, whose userid is *name*, has changed the signon parameters for the given data source.

OPER: Default password changed from console, userid = '*name*'

This message indicates that the operator, whose userid is *name*, has changed the default password from the Client Console. The default password is used for a new userid or after a userid's password is reset.

OPER: Existing Data Source *src* added from console, userid = '*name*'

This message indicates that the operator, whose userid is *name*, has added the given data source, which already exists but it not known to the service.

OPER: Password for userid '*name1*' reset from console, userid = '*name2*'

This message indicates that the operator, whose userid is *name2*, has reset the password for the userid *name1*. When a password is reset, it will assume the value of the default password. It is subject to the rule that it must be changed at first use when this option is enabled. This provides an easy way of dealing with a user who does not remember his password.

OPER: Password security bits and parameters set to 0x*nn*, *aa*, *bb*, *cc* from console, userid = '*name2*'

This message indicates that the operator, whose userid is *name*, has changed the password security bits and password security parameters from the Client Console. The bit mask *nn* determines what constitutes a legal password and whether or not a new user needs to change his password on first use. The value *aa*, *bb* and *cc* are the minimum password length, the maximum signon retry count, and the length of time for which a user gets locked out after he exceeds the maximum number of signon retries.

OPER: Permissions for userid '*name1*' updated from console, userid = '*name2*'

This message indicates that the operator, whose userid is *name2*, has customized the permissions for the userid *name1*.

OPER: {Service | Daemon} Data Source *srcname* parameters modified from console, userid = '*name*'

This message is one of a series of messages that are designed to allow auditing of all operator actions. Such messages always start with the string "OPER:", thus making it easy to find them by searching for this string in the log file. They also specify the userid of the operator that performed the action.

This message indicates that the operator, whose userid is *name*, has changed one of the parameters of the data source *srcname*.

OPER: {Service | Daemon} debug mode {enabled | disabled} from console, userid = '*name*'

This message indicates that the operator, whose userid is *name*, has enabled or disabled debug mode for the service. You should almost never need to do this, except when you are having trouble getting the service to run. Besides the log file, you should also look at the Windows Application Event Log, as the Client service also writes some message in the Event Log if it encounters problems during startup.

OPER: {Service | Daemon} enable\_status\_file set to *dd* from console, userid = '*name*'



(Clustered Windows systems only) This message indicates that the operator, whose userid is *name*, has changed the value of the parameter *enable\_status\_file*. If this parameter is set to True, the service maintains a status file, which contains information about the state of the various data sources it controls. This file is named "dbstatus.cfg" and it resides in the config sub-directory. It is used to restart runs that were active before the service is restarted. The difference between using this method and setting *run\_at\_startup* to True for a data source, is that latter causes the run to always be started, even if it was not active when the service was taken down. If you attempt to set this parameter to true on a system that is not clustered, the parameter will be ignored.

OPER: {Service | Daemon} listen port set to *dddd* from console, userid = '*name*'

This message indicates that the operator, whose userid is *name*, has changed the port number on which the service listens. The service starts out by using the port number that is defined at install time, after that you can change it from the console. But if you do so, the change will only take effect after you stop and restart the service.

OPER: {Service | Daemon} log parameters updated from console, userid = '*name*'

This message indicates that the operator, whose userid is *name*, has changed some of the logging parameters for the service.

OPER: {Service | Daemon} log file switched from console, userid = '*name*'

This message indicates that the operator, whose userid is *name*, has made the service switch log files. The current log file is closed and a new one with a different name is opened. The new log file will use today's today date as part of the name, unless the old file was already doing so, in which case it will also add the time to the file name to make it unique.

OPER: {Service | Daemon} n\_script\_threads set to *dd* from console, userid = '*name*'

This message indicates that the operator, whose userid is *name*, has changed the number of threads for handling scripts that are launched by the service.

OPER: {Service | Daemon} startup\_delay set to *ddd* from console, userid = '*name*'

This message indicates that the operator, whose userid is *name*, has changed the *startup\_delay* parameter in the service's configuration file.

OPER: {Service | Daemon} trace mask set to *dd* from console, userid = '*name*'

This message indicates that the operator, whose userid is *name*, has modified the trace mask, which causes tracing of Console and/or IPC traffic to be enabled or disabled, based on the value specified.

OPER: Script "*filename*" launched from console, userid = '*name*'

This message indicates that the operator, whose userid is *name*, has launched the given script file from the Client Console.

OPER: Userid '*name1*' added from console, userid = '*name2*'

This message indicates that the operator, whose userid is *name2*, has added userid *name1* to the service's configuration file.

OPER: Userid '*name1*' removed from console, userid = '*name2*'

This message indicates that the operator, whose userid is *name2*, has removed userid *name1* from the service's configuration file.

OPER: Userid '*name1*' unlocked from console, userid = '*name2*'

This message indicates that the operator, whose userid is *name2*, has unlocked userid *name1* which was locked out because of too many consecutive bad signons. Locked out userids will automatically get unlocked when the lockout period expires. This command allows the operator to make this happen immediately.

*RC\_rpc\_name*

This message, which is logged when the debug option is enabled, identifies the RPC being executed. This is intended to be used in conjunction with a service trace.

Removing data source *name*

In some cases, the removal of a data source and its directories must be deferred until the spawned DBCIntCfgServer run terminates. In those cases, this message appears when the process of removing the data source begins.

Scheduling for data source name delayed by *ss* seconds due to *blackout\_period*

This message indicate that the service cannot start a scheduled run because there is a blackout period in effect. The service simply postpones the start time of the run to coincide with the end the blackout period.

{Service | Daemon} Log file switched to "*filename*" (*reason*)

This message is logged when the log file is about to be switched. It contains the new filename and the reason for the switch, which includes Operator Keyin, Max File Size, and Date Change.

Start\_of\_run: data\_source = *srcname*, process\_id = *dddd*, command = *name*, originator = who[ (*sched\_type*) ]

This message is logged whenever the service launches a Client run for a data source. The command name is *process* or *clone* for DBClient runs and *manage*, *configure*, *verifysource*, *redefine* or *generate* for DBCIntCfgServer runs.

The launched run can be initiated by any of the following: *scheduler*, *select*, *console*, *config*, *verify*, *auto-redefine* or *auto-generate*. If the run is started by *scheduler*, the *sched\_type* value defines the type of scheduling used for the client run. Possible values for *sched\_type* include: *daily*, *fixed delay*, *err\_retry* or *startup*.

Starting scheduled process command for data source *name*

This message is logged when the service launches a process command as a result of scheduling.

Starting Session *dd*, socket = *nnn*, IP address = *ipaddr*

This message is logged when a connect request is successfully handled. These requests include Console, Client Configurator connections and IPC connections from spawned runs.

Starting *verifysource* command for data source *name*

When the service adds a data source, it launches DBCIntCfgServer to execute a *verifysource* command. This command determines whether the database and server access parameters are valid. If they are invalid, the Console operator must correct them.

Too many consecutive bad signon attempts for CNS Session *nn* (userid = *name*), session terminated

This message indicates that the given userid failed to signon after the maximum number of retries, which is currently 3. You should never see this message when using the Client console, as it disconnects after a failed signon.

Userid '*name*' locked after *nnn* bad signons

This message indicates that the given userid failed to sign on after the maximum number of retries, which is currently 3. As a result the userid is locked to protect against password guessing.

Userid '*name*' unlocked after *ddd* seconds

This message indicates that the given userid is automatically unlocked after the lockout timeout period expires.

## Error Messages

The following messages are only written to the Client Manager Service log file, which is located in the `logs` directory in the global working directory. These messages are prefixed by the timestamp *hh:mm:ss* (where *hh* is hours, *mm* is minutes, and *ss* is seconds)

ERROR: Add\_DataSource: `chdir` failed for working directory "*wdir*", *errno=number* (*errortext*)

When adding a data source the service creates the working directory for the data source and its subdirectories. It then updates the service's configuration file and changes the current directory to the data source's working directory and creates the client's configuration file. After it creates the client's configuration file, it switches back to the service's (or global) working directory. This message is used to indicate that the attempt to change the program's current directory failed. The directory *wdir* is the target directory for which the `chdir` command failed. The system error number and associated text should help determine why the operation failed.

ERROR: Add\_ExistingDS: `chdir` failed for directory "*dir*", *errno=number* (*errortext*)

When adding an existing data source the service changes its current directory to the data source's working directory and verifies that its subdirectories exist. When done it changes the current directory back to the service's working directory. This message is used to indicate that the attempt to change the program's current directory failed. The directory *wdir* is the target directory for which the `chdir` command failed. The system error number and associated text should help determine why the operation failed.

ERROR: Add\_ExistingDS: `chdir` failed for working directory "*wdir*", *errno=number* (*errortext*)

When adding an existing data source, the service uses a series of `chdir` commands to determine if the Client's global working directory contains the required subdirectories. This message is used to indicate that the attempt to change the program's current directory failed, *wdir* is the target subdirectory for which the `chdir` command failed. The system error number and associated text should help determine why the operation failed. The most likely cause is that the target directory does not exist.

ERROR: Bad input line '*text\_line*' in file "globalprofile.ini"

(UNIX only) This message indicates that the file `globalprofile.ini` has a bad line that is displayed. This file should be created starting from the sample file that is provided with the release. If you are careful when editing it and just change the values of the parameters you will never see this error.

ERROR: Bad section header '*text\_line*' in file "globalprofile.ini"

(UNIX only) The only section header used in the file `globalprofile.ini` is `[dbridge]`. This file should be created starting from the sample file that is provided with the release. If you are careful when editing it and just change the values of the parameters you will never see this error.

ERROR: Bad user id '`userid`' specified in file "`globalprofile.ini`"

(UNIX only) This message indicates that the `userid` provided on the line "`userid = name`", is not recognized as being a valid user-id.

ERROR: Binary configuration file "`name`" has a bad checksum

Binary configuration files use a checksum to ensure that the file is not corrupt. Do not try to use a binary editor to update a binary configuration file. If you change even a single byte, the checksum will be wrong. You should use the Console, the Client Configurator or the `dbctrlconfigure` utility (`export/import` commands) to update the service's configuration file. The most likely reason for this error would be that the file was patched or the file is corrupt.

ERROR: Binary configuration file "`name`" is corrupt -- data source number `nn` has a name of ""

ERROR: Binary configuration file "`name`" is corrupt -- data source `source` has no working directory

ERROR: Binary configuration file "`name`" is corrupt -- data source `source` has no client directory

These three messages indicate that the binary configuration is corrupt (this should not happen under normal circumstances).

If the service will not start, check the service's log file to see if you got one of these messages. If so, copy the backup configuration file "`dbcontrol.bak`" as "`dbcontrol.cfg`" to get around this problem. You will want to first export the file to see if it is missing a data source. If this is the case, add this back in by an "Add Existing Data Source" from the client console, or by editing the file and importing it.

The file "`dbcontrol.bak`" gets created in the service's config directory when a data source is added or removed. For all other actions that cause the configuration file to be updated (such as disabling a data source), the service does not copy the file "`dbcontrol.cfg`" as "`dbcontrol.bak`" before updating it.

ERROR: Binary configuration file "`name`" has the wrong check word

The service only handles binary configuration files; such files start with an 8-byte pattern that is used to determine if the file is binary or not. If the pattern does not match this message is logged. The most likely reason for this would be that the file is not a binary file.

ERROR: Binary status file "`name`" has a bad checksum

(Clustered Windows systems only) In addition to the configuration file "`dbcontrol.cfg`" the service uses the binary file "`dbstatus.cfg`" to keep track of the state of the various data sources. This file resides in the config sub-directory. This is particularly useful in a clustered architecture when a node switch occurs. It allows the service to restart the process commands that were running before the node switch occurred. This file uses a checksum to ensure that it has not been corrupted. If the checksum is bad, this message is displayed. Do not edit this file under any circumstances. If you get this error, simply delete the file from the config sub-directory. This will simply make the service forget what it was doing prior to being forcibly shutdown and it will revert to starting up normally.

ERROR: Binary status file "`name`" has the wrong check word

(Clustered Windows systems only) See previous message for details on the file "dbstatus.cfg". Binary configuration files use a checksum to ensure that the file is not corrupt. Do not try to use a binary editor to update a binary configuration file. If you change even a single byte, the checksum will be wrong.

ERROR: chdir failed for working directory "*name*", *errno=number* (*errortext*)

During normal operations, the service uses the `chdir` command to change the current directory. This error indicates that the service was unable to switch the service's (global) working directory to the data source's working directory or vice versa. The path of the target directory is specified by *name*.

ERROR: Client configuration file "*name*" has bad checksum

If you use the service, you must use binary files for all Client configuration files. Binary configuration files start with an 8-byte pattern. If a Client configuration file doesn't start with this pattern, or if the binary file has been edited and the checksum corrupted, this message occurs.

ERROR: Client configuration file "*name*" is corrupt, *errno=number* (*errortext*)

This message indicates that the binary configuration file was edited using a binary editor or patched and was corrupted as a result. Using binary editors to update a binary configuration file should be strictly avoided altering a single byte will render the checksum as wrong. To update the client configuration files, use the available options in the Console and the Client Configurator, or use the `dbutility export/import` commands if you need to make manual edits.

ERROR: Close failed for binary configuration file "*name*", *errno=number* (*errortext*)

This error indicates that an I/O occurred when the service tried to close its binary configuration file after rewriting it. To determine why the error occurred, refer to the system error number and associated text.

ERROR: Close failed for client configuration file "*name*", *errno=number* (*errortext*)

This error indicates that an I/O occurred when the service tried to close the client's binary configuration file after rewriting it. The system error number and associated text should help in determining why the close failed.

ERROR: CreateFile failed for file "*name*", *error=number* (*errortext*)

(Windows only) The service uses the `CreateFile` library procedure to create new files when file security is enabled. This error indicates that the file could not be created. The service will create the file using default security instead. The supplied error number and corresponding text should indicate why the failure occurred.

ERROR: CreateMutex failed for '*name*', *error=number* (*errortext*)

(Windows only) This message, which can occur during the startup of the service, indicates that a Windows internal error has occurred while attempting to create a mutex resource. If this error occurs, contact Micro Focus Technical Support.

ERROR: Createprocess failed for client run, *error=number* (*errortext*)

(Windows only) This message, which can occur when the service launches a run of DBClient or DBCntCfgServer, indicates that the attempt to create the process failed. The system error number and associated text should help determine the reason for the failure. This error can occur if edit the service configuration file and provide the wrong directory for the Client. We recommend that you update the service configuration file via the Client Console or via the Migrate program if you're upgrading.

ERROR: CreateSemaphore failed for 'script\_work\_queue\_sem', error=*number* (*errortext*)

(Windows only) This message, which can occur during the startup of the service, indicates that a Windows internal error occurred while attempting to create a semaphore resource used for signaling the script execution thread. If this error occurs, contact Micro Focus Technical Support

ERROR: CreateThread failed for "*name*", error=*number* (*errortext*)

(Windows only) This message indicates that a Windows internal error occurred while trying to create a thread. The program uses threads to launch scripts as they can take a long time to execute. To determine the reason for the failure, refer to the system error number and associated text.

ERROR: Execute\_Command returned *status* -- killing RC session

This message indicates that the service encountered a problem when trying to execute an RPC received from the Client Console or the Client Configurator. The error is fatal and causes the session to terminate. If this error causes the Client Console to hang, restart the Client Console and then reconnect to the service. If the problem persists make sure that you are running a version of the Client Console that is compatible with the service.

ERROR: Execute\_IPC\_Response returned *status*

This message indicates that the service encountered an error while processing the response to an RPC that had been sent to a spawned client. The message is usually preceded by another message that provides information about why the failure occurred.

ERROR: Get\_IPC\_Response(): seq\_no validation error; send\_no=*hhhh*, cnt= *hhhh*,  
recv\_no= *hhhh*

This message indicates that an internal error occurred when processing a response to an RPC that had been sent to a spawned client. For the service to correctly identify and process an RPC response, the response must have the same sequence number as the RPC command. If the response has a different sequence number, this message is logged and an RPC error status is returned to the Client Console or the Client Configurator.

ERROR: getcwd failed, errno=*number* (*errortext*)

The service uses the `getcwd` command to get the current directory. If the command fails this message is logged. The system error number and associated text that is provided should help determine the reason why this operation failed.

ERROR: Incomplete length word (*dd* bytes) received for Session *nn*, closing connection

This message is displayed when the service gets a non-Databridge connection that sends it less than 4 characters. The service expects to get a 4-byte length at the start the first message from the Client Console or a launched run. When it gets less than 4 characters, it logs this message and closes the connection, instead of going back and trying to read the remaining characters, which could hang the run until the connection is closed.

ERROR: Initialization for Session *dd* did not complete in *nnn* seconds, terminating connection

This message indicates that a new connection to the service failed to execute the necessary RPCs to setup the session within the expected length of time. This timeout is designed to prevent bogus connections from sitting around tying up resources.

ERROR: IPC session *nn* is not properly setup, session terminated

This message indicates that a spawned client provided a bad data source name in the signon RPC data after it successfully connected to the service. This should never happen; if it does contact Micro Focus Technical Support.

ERROR: Launch of client failed, *errno=number (errortext)* -- see file "fork.log" for clues

(UNIX only) This message indicates that the daemon was unable to launch the Client either when starting a run of DBClient or DBCIntCfgServer. This error can occur if you enter the wrong directory for the Client when editing service configuration file. We recommend that you update the service configuration file by using the Client Console; or when upgrading, the migrate utility. To determine why the operation failed, refer to the system error number and associated text, and to the file `fork.log`. The daemon create this file in its working directory when this error occurs.

ERROR: Launched run for data source *name* did not connect back to service within *ddd* seconds

This message indicates that a run launched by the service failed to connect to the service and execute the necessary RPCs to setup the IPC connection. The most likely cause for this error is that the launched run encountered an error during startup and terminated prematurely. The 6.6 service implemented a timeout that allows the service to free the resources associated with the run and to send a response to the console that prevents it from hanging.

ERROR: Message length *0xhhhhhhh* exceeds limit -- closing connection

This message is displayed when the service gets a message whose length is too long to be legitimate. This allows the service to protect itself against rogue connections.

ERROR: Message length *0xhhhhhhh* for RC\_Initialize RPC is incorrect -- closing connection

This message is displayed when the service gets a non-Databridge connection that sends it an unrecognizable message. The service knows the length of the initial message it gets from the Client Console or a launched run. When this length is wrong, it logs this message and closes the connection.

ERROR: Open failed for client configuration file "*name*", *errno=number (errortext)*

This message indicates that the service received an error when it tried to open an existing client file configuration file during an `add data source` command. To determine why the operation failed, refer to the system error number and associated text.

ERROR: Open failed for file "*name*", *errno=number (errortext)*

This message indicates that the service received an error when it tried to open a file. To determine why the operation failed, refer to the system error number and associated text.

ERROR: Open failed for global configuration file "globalprofile.ini"

(UNIX only) This message indicates that the program was unable to access the file `globalprofile.ini` in the directory `/etc/Attachmate/DATABridge/6.6`. This file must be created before attempting to do any Client operations.

ERROR: `pthread_create` failed for '*name*', *error=number* (*errortext*)

(UNIX only) This message, which can occur during the startup of the daemon, indicates that a system error occurred when a thread was being created. The program uses threads to launch scripts as they can take a long time to execute. To determine the reason for the failure, refer to the system error number and associated text.

ERROR: RC session no longer exists, destination index *nn* for CNS Session *nn*

This message indicates that an IPC response for a Console or Client Configurator session was received, but an attempt to find the matching session failed because the session no longer exist. This error is not fatal; the response is simply discarded.

ERROR: Read failed for binary configuration file "*name*", *errno=number* (*errortext*)

This message indicates that the program was unable to read the service binary configuration file `dbcontrol.cfg`. To determine the reason for the failure, refer to the system error number and associated text.

ERROR: Read failed for binary status file "*name*", *errno=number* (*errortext*)

(Clustered Windows systems only) This message indicates that the program was unable to read the service binary status file `dbstatus.cfg`. To determine the reason for the failure, refer to the system error number and associated text.

ERROR: Read failed for client configuration file "*name*", *errno=number* (*errortext*)

This message indicates that the client's binary configuration file `dbridge.cfg` could not be read. To determine the reason for the failure, refer to the system error number and associated text.

ERROR: Read failed for file "*name*", *errno=number* (*errortext*)

This message indicates that the attempt to read the log file descriptor from the file `log.cfg` failed. To determine the reason for the failure, refer to the system error number and associated text.

ERROR: Select failed, *error=number* (*errortext*)

This message indicates that a sockets select call returned a count less than zero. The error is ignored and the operation is retried after a brief delay.

ERROR: `sem_init` failed for '*script\_work\_queue\_sem*', *error=number* (*errortext*)

(UNIX only) This message, which can occur during the startup of the service, indicates that an internal error occurred while attempting to create a semaphore resource used for signaling the script execution thread. If this error occurs, contact Micro Focus Technical Support.

ERROR: system command failed for script "*filename*", *errno=number* (*errortext*)

This message indicates that the service was unable to launch a script file associated with a data source, either when a run terminated with a non-zero exit status or when a request to launch a script was received. Either the script is in error or the script file doesn't exist.

ERROR: Unable to allocate *nnnn* bytes of memory



This message indicates that the operating system cannot allocate the specified block of storage because there isn't enough installed memory.

ERROR: Unable to create client configuration file "*name*", *errno=number* (*errortext*)

This message indicates that an error occurred when the service tried to open and write to the client binary configuration file. To determine why the operation failed, refer to the system error number and associated text.

ERROR: Unable to create working directory for the daemon

(UNIX only) This message can occur if the daemon is started from the command line under a user ID that doesn't have the necessary privileges to create the daemon's working directory. Typically this message would not occur; the scripts that launch the daemon require that the working directory be created in advance.

ERROR: Unable to expand block to *ddd* bytes

This message indicates that the operating system is unable to expand an allocated block of memory, either because there isn't enough installed memory.

ERROR: Unexpected RPC response received for index *dd* of {CNS | IPC} Session *nn*

This message indicates that the service received an unexpected RPC response. If the problem persists, contact Micro Focus Technical Support.

ERROR: Unimplemented command *rpcnumber*

This message indicates that the Console or Client Configurator sent the service an RPC whose number is not known by the service. Unless you have you are using a service that is older than the Console, you should never get this error.

ERROR: Unknown session type *dd* encountered in Kill\_Session()

This message indicates that an internal error occurred caused by an illegal type in a session. If this error occurs, contact Micro Focus Technical Support.

ERROR: Value out of range for umask in file "globalprofile.ini"

(UNIX only) This message indicates that the umask line in the file gloabalprofile.ini is bad. This value needs to be an octal number in the range 0 to 0777. It is AND'ed with the default security bits to set the security for a file that is created, just like umask does in UNIX.

ERROR: Write failed for binary configuration file "*name*", *errno=number* (*errortext*)

This message indicates that the program was unable to write the service binary configuration file *dbcontrol.cfg*. The system error number and associated text should help determine the reason for the failure.

ERROR: Write failed for client configuration file "*name*", *errno=number* (*errortext*)

This message indicates that the program was attempting to create the Client binary configuration file *dbridge.cfg*. To determine the reason for the failure, refer to the system error number and associated text.

ERROR: Write failed for file "*name*", *errno=number* (*errortext*)

This message indicates that the program was unable to create the log descriptor file *log.cfg*.

ERROR: XDR\_Copy failed; RC\_Sess\_buflen = *0xhhhh*, IPC\_Sess\_buflen = *0xhhhh*

This message indicates that an internal error occurred when the RPC data from a console session run is copied to a send buffer. If this error occurs, contact Micro Focus Technical Support.

ERROR: XDR receive failure for sockettype socket *nnn*

This message indicates that the attempt to read and decode the data for an RPC failed for the specified socket type. The socket type can be `CONSOLE`, `IPC`, or `LIMBO`. `LIMBO` indicates that the connection type is not yet known. This error may occur when the connection closes.

ERROR: You must be 'su' or userid 'userid' to run the daemon

(UNIX only) This message indicates that the script used to run the service uses a different userid than the one specified in the file `globalprofile.ini`. To switch the service to a user other than that in `globalprofile.ini`, start the service after you start a run as the root user.

SOCKETS ERROR: accept call failed for remote console, error=*number* (*errortext*)

This message indicates that the service got an error when it tried to issue an accept in response to a connect request. This is a network error; the associated error number and error text may help you determine why it occurred.

SOCKETS ERROR: bind call failed for remote console TCP port number *nnn*, error=*number* (*errortext*)

This message indicates that the service was unable to bind the `ipc_port` to the given socket. To resolve this problem specify a different port for the `ipc_port` parameter in the service configuration file. This error usually occurs when the service starts.

SOCKETS ERROR: getaddrinfo call failed, error=*number* (*errortext*)

This message appear when the service start up, this is call is used when trying to figure what type of listen requests (IPV4, IPV6 or both) need to be handled. Contact Micro Focus Technical Support if you get this error.

SOCKETS ERROR: listen call failed for remote console, error=*number* (*errortext*)

This message indicates that the service was unable to listen for incoming calls on the specified port. To resolve this problem, specify a different port for the `ipc_port` parameter in the service configuration file. This error usually occurs when the service starts.

SOCKETS ERROR: recv call failed for {CNS | IPC} Session *nn*, error=*number* (*errortext*)

This message indicates that the service encountered an error while receiving data. This typically occurs when the connection is reset.

SOCKETS ERROR: recv call for {CNS | IPC} Session *nn* detected a closed connection

This message indicates that the service detected a closed connection while receiving data.

SOCKETS ERROR: send call failed for {CNS | IPC} Session *nn*, error=*number* (*errortext*)

This message indicates that the service encountered an error while sending data. This typically occurs when the connection is closed or reset.

SOCKETS ERROR: setsockopt(SO\_REUSEADDR) call failed for remote console TCP port, error=*number* (*errortext*)

This message, which can occur when the service starts up, indicates that the attempt to set the specified socket option failed.

SOCKETS ERROR: socket call failed for remote console TCP port, error=*number*  
(*errortext*)

This message, which can occur when the service starts up, indicates that the attempt to set the specified socket option failed.

SOCKETS ERROR: socket call failed for remote console TCP port, error=*number*  
(*errortext*)

This message indicates that the system is out of sockets. Reboot the system.

SOCKETS ERROR: WSStartup error *nnn*

(Windows Only) This message indicates that the attempt to initialize Windows sockets failed. The only thing to do is to reboot and hope that the problem goes away.

## Windows Application Log Messages

CheckTokenMembership() failed for group '*name*', error=*errno* (*errortext*)

When file security is enabled and you specify group names for the Databridge Administrators the service uses this system call to determine if the current user is a member of this group. This is an indication that you are not using the SYSTEM ACCOUNT or the account specified as UserID in the Windows Registry, and that the attempt to verify the group membership of the current account in the Databridge Administrator group failed. We recommend that you always run the service using the default account, which is the SYSTEM ACCOUNT.

CreateDirectory() failed for "*path*", error=*errno* (*errortext*)

This message indicates that the service was unable to create the directory specified by path. This needs to be rectified before the service can run effectively. The most likely cause for this is that you entered a bad file name when running the updatepath utility at install time. You need to use regedit to correct this.

End cmd command for data source *name*, process\_id= *pid*, exit\_code= *nnnn*  
(*exit\_code\_text*)

This message is written to the Application Log when a client run launched by the service terminates.

*cmd* is the client command, which can be *clone*, *configure*, *generate*, *process* or *redefine*. Other client commands such as *manage* and *verifysource* are not logged. The process ID is used to find the matching entry when the client run is launched. *nnnn* (*text*) is the client exit code and the associated explanatory text that is written to the client log file at the end of the run.

Error in processing binary configuration file "*dbcontrol.cfg*"

This message is written to the Application Log when the service fails to start because it cannot read the configuration file. Looking at the application event log is a good place to start when the service will not start.

Failed to create configuration file "*name*"

The first time you run the service it creates its global working directory and the subdirectories *config*, *logs* and *locks*. It then creates the binary configuration file *dbcontrol.cfg* in the *config* subdirectory. This message indicates that an error occurred while writing this file.

Failed to open configuration file "*name*"

This message indicates that the service was unable to open the configuration file and that the subsequent attempt to create or verify the existence of the working directory failed.

INSTALLDIR parameter is an empty string in the Windows registry

The service uses the INSTALLDIR string in the Windows registry key for the key for the Databridge client. This string is created by the installer. This message indicates that the string has been wiped out. If things are that messed up, you probably should reinstall the Databridge client

Launch *name* command for data source *name* process\_id=*dddd*, started by *who* [*sched\_type*]

This message is written to the application log when a client run is launched by the service.

*cmd* is the client command, which can be *clone*, *configure*, *generate*, *process* or *redefine*. Other client commands such as *manage* and *verifysource* are not logged. The process ID is used to find the matching entry when the client run is launched. *who* [*sched\_type*] indicates how the run was launched. Possible values include *auto-generate*, *auto-redefine*, *console*, *config* and *scheduler*. Runs started by the scheduler also the scheduling type, which can be *daily*, *fixed\_delay*, *err\_retry* or *startup*. *startup* is a result of having the *run\_at\_startup* parameter set to *True* for the data source. *daily* and *fixed\_delay* indicate that the run was scheduled because the corresponding type of scheduling is enabled for the data source.

*err\_retry* indicates that a process command was restarted because the service determined that retrying was appropriate. Most errors for which the service retries are controlled by the *max\_retries* and *sched\_retry\_secs* parameters. Exit codes that are related to connectivity and inability to connect to the database do not use these parameters. Instead they retry forever using longer and longer intervals until they reach the maximum delay time (currently 5 minutes).

Launch of *name* command for data source *name* failed

This message is written to the application when the launch of a client run fails.

Set working directory failed

This message indicates that the service was unable to change the current directory to the working directory. The most likely cause for this is that you entered a bad file name when running the *updatepath* utility at install time. You need to use *regedit* to correct this.

SHCreateDirectoryEx() failed for "*path*", error=*errno* (*errortext*)

This message indicates that the service was unable to create the working directory specified by *path*. This needs to be rectified before the service can run. The most likely cause for this is that you entered a bad file name when running the *updatepath* utility at install time. You need to use *regedit* to correct this. Or you do not have write access to the directory in which you trying to create the working directory.

Started

When the service is successfully started it writes this message to the Application Log.

Unable to create DACL

When file security is enabled, the service creates a DACL that it uses to set the file security for all directories and files it creates. If it is unable to create it will revert to using default security, it writes this message to the Application Log as a reminder that something is not working correctly. The most likely cause of this error is that you are not running the service using the proper account. We recommend that you always run the service using the default account, which is the SYSTEM ACCOUNT

Unable to read *reg\_str* parameter from Windows registry (result = *nn*)

This message indicates that the service was unable to retrieve the value of the string *reg\_str* from the Windows registry key for the Databridge Client, where *reg\_str* is one of the following names: FileSecurity, INSTALLDIR, UserID or WORKINGDIR. Use regedit to see if the Windows registry entry is corrupted. You probably should reinstall the Databridge Client software.

User '*name*' not authorized to run Databridge client

When file security is enabled, the service writes this message to the Application Log if it determines that the current userid is not the SYSTEM account or a userid that has access to the Databridge working directory files. We recommend that you always run the service using the default account, which is the SYSTEM account.

## Client Console and Configurator Messages

The following messages only occur when using the Client Console or the Client Configurator and are not related to Client Console errors, which indicate a problem between the Client Console and the Client service.

3000 - Success with info  
3001 - Console must first issue an RC\_Initialize function  
3002 - Session not signed on  
3003 - Signon failed  
3004 - Error decoding RPC parameters  
3005 - Illegal function  
3006 - Databridge client running a process command, request rejected  
3007 - Databridge client running a clone command for one or more of the selected data sets  
3008 - Databridge configuration server already running for the data source, request rejected  
3009 - Maximum number of attachments for the session or the data source exceeded, request rejected  
3010 - Data source not found  
3011 - An error occurred while spawning the Databridge client  
3012 - Session not attached to the data source, request rejected  
3013 - An RPC is pending, request rejected -- try again in a few seconds  
3014 - The specified run cannot be found  
3015 - An error occurred while attempting to switch the log file  
3016 - An illegal trace mask was specified in the RC\_SetParameter RPC  
3017 - An invalid parameter number was specified in the RC\_SetParameter or RC\_GetParameter RPC  
3018 - No host variables  
3019 - No statistics  
3020 - Statistics not yet available; issue the request after the client initialization is completed  
3021 - The data source is disabled, you must enable it before this function can be performed

3022 - The specified process cannot be found  
 3023 - The session is already attached to the specified run  
 3024 - Proc not found  
 3025 - Manager running  
 3026 - Manager closed  
 3027 - Data source not selected  
 3028 - Relational data base error  
 3029 - Data source locked, request rejected  
 3030 - Error loading the client control tables  
 3031 - You cannot specify partial state information; AFN, ABSN, SEG & INX  
 must all be specified  
 3032 - Requested function is not supported  
 3033 - Internal error  
 3034 - Error while encoding RPC data  
 3035 - Malformed RPC  
 3036 - Wrong Password  
 3037 - Maximum number of data sources reached  
 3038 - 3050 (Currently unused)  
 3051 - Internal error, list is empty  
 3052 - Client internal error  
 3053 - Error while attempting to update client control tables  
 3054 - No client control tables found in the relational database  
 3055 - Client control table version mismatch -- run the dbfixup utility to  
 fix this situation  
 3056 - You do not have the proper credentials to run this RPC  
 3057 - WARNING: Data source already selected -- request ignored  
 3058 - Error while attempting to create the client control tables  
 3059 - Data source already exists  
 3060 - DBClntCfgServer unable to connect to server -- verify hostname,  
 hostport parameters for data source  
 3061 - Databridge client error  
 3062 - Backup error  
 3063 - Illegal update  
 3064 - Set not found  
 3065 - DATATABLES entry not found  
 3066 - DMS\_ITEMS or DATAITEMS entry not found  
 3067 - Memory allocation error  
 3068 - Invalid layout  
 3069 - Rollback error  
 3070 - DMSII to relational database mapping failed  
 3071 - Invalid backup  
 3072 - Invalid handle  
 3073 - Audit File processing only  
 3074 - Parameter value out of range  
 3075 - Cannot connect to the server  
 3076 - Not implemented  
 3077 - Invalid parameter Value for RPC  
 3078 - Attempt to create directory failed  
 3079 - File IO error  
 3080 - File open error  
 3081 - File creation error  
 3082 - File 'dbcontrol.cfg' in error, unable to update it  
 3083 - Data source cannot be removed it is currently active  
 3084 - Files missing in data source's dbscripts directory  
 3085 - Client configuration file in error

3086 - Invalid data source name  
3087 - Redefine command failed  
3088 - Illegal RPC for current DBConfig state  
3089 - This RPC is only allowed for DBConfig  
3090 - Illegal Option  
3091 - Option has bad argument  
3092 - Default client configuration file has the wrong version  
3093 - Default client configuration file is corrupt  
3094 - Attempt to change directory failed  
3095 - Databridge client running a redefine command, request rejected  
3096 - Databridge client running a generate command, request rejected  
3097 - Command line too long  
3098 - You need to run dbscriptfixup before you can customize control table using DBConfig  
3099 - Name already in use by another object in the database  
3100 - Databridge client is already running a script for the data source, request rejected  
3101 - Cannot perform add as userid that already exists  
3102 - Cannot remove a userid that does not exist  
3103 - Maximum number of userids reached, cannot add another userid  
3104 - Cannot remove the last userid  
3105 - Script quota exceeded  
3106 - Cannot remove a userid that is active  
3107 - Userid locked

