Oracle® Fusion Middleware
Logdump Reference for Oracle GoldenGate
12c (12.2.0.1)
E66976-03

July 2016
Documentation for system administrators and database administrators that describes how to work with the Logdump trail-reader utility
## Contents

Preface ............................................................................................................................................................... vii  
Audience ...................................................................................................................................................... vii  
Documentation Accessibility .................................................................................................................... vii  
Related Information.................................................................................................................................... vii  
Conventions................................................................................................................................................ viii

### 1 Using the Logdump Utility

1.1 Getting Started with Logdump........................................................................................................... 1-1  
1.2 Viewing the First Record ................................................................................................................ 1-1  
  1.2.1 Running Logdump........................................................................................................... 1-1  
  1.2.2 Setting Up the View ....................................................................................................... 1-1  
  1.2.3 Opening a Trail File....................................................................................................... 1-2  
1.3 Executing Basic Logdump Tasks..................................................................................................... 1-3  
  1.3.1 Finding the Next Good Record Header ........................................................................... 1-3  
  1.3.2 Finding the Beginning, Middle, and End of a Transaction ........................................... 1-3  
  1.3.3 Scanning for the End of a Transaction .......................................................................... 1-4  
  1.3.4 Going to a Specific RBA in the File .............................................................................. 1-4  
  1.3.5 Filtering Based on a Table or Data File Name................................................................. 1-5  
  1.3.6 Removing the Current Filter Criteria ............................................................................ 1-5  
  1.3.7 Filtering on Multiple Conditions .................................................................................... 1-5  
  1.3.8 Count the Records in a Trail File..................................................................................... 1-5  
  1.3.9 Saving Records to a New Trail File.................................................................................. 1-5  
  1.3.10 Closing the Current File and Opening the Next One in the Trail .............................. 1-6  
  1.3.11 Keeping a Log of Your Session...................................................................................... 1-6  
  1.3.12 Seeing the Current Logdump Environment.................................................................... 1-6  
  1.3.13 Getting Online Command Help.................................................................................... 1-6  
  1.3.14 To Exit Logdump........................................................................................................... 1-6  
1.4 Evaluating Transaction Size............................................................................................................. 1-6  
1.5 Maintaining Command History ....................................................................................................... 1-7  
1.6 Viewing Logdump Command Information and Syntax...................................................................... 1-7
2 Logdump Commands

2.1 Logdump Command Summary ................................................................. 2-1
2.2 ARGETNONDATACHANGES | ARSTOPNONDATACHANGES ..................... 2-4
2.3 ASCIIDATA | EBCDICDATA ................................................................. 2-5
2.4 FMW Generic Topic ............................................................................ 2-5
2.5 FMW Generic Topic ............................................................................ 2-5
2.6 BEGIN .................................................................................... 2-5
2.7 BULKIOSTATS ............................................................................ 2-6
2.8 CALCTLFKEY ........................................................................... 2-6
2.9 CD .................................................................................................. 2-6
2.10 COMPUTETIMESTAMP ................................................................. 2-6
2.11 COUNT .................................................................................... 2-7
2.12 CTIME ..................................................................................... 2-11
2.13 DATAFILE ............................................................................. 2-12
2.14 DEBUG .................................................................................... 2-12
2.15 DECRYPT ................................................................................ 2-12
2.16 DETAIL ................................................................................... 2-13
2.17 DUMP ....................................................................................... 2-14
2.18 ENCRYPT ................................................................................ 2-14
2.19 ENV .......................................................................................... 2-15
2.20 ESBLOCK ............................................................................... 2-16
2.21 EXIT ....................................................................................... 2-16
2.22 FC ............................................................................................ 2-16
2.23 FILEHEADER ........................................................................ 2-18
2.24 FILES ........................................................................................ 2-25
2.25 FILTER ..................................................................................... 2-26
2.26 FLOAT ..................................................................................... 2-32
2.27 GGSAUDITREAD .................................................................. 2-33
2.28 GGSTOKEN ........................................................................... 2-34
2.29 GGSTOKEN ........................................................................... 2-35
2.30 GHDR ...................................................................................... 2-36
2.31 HASHCLEAR ........................................................................ 2-36
2.32 HASHSTATS ........................................................................... 2-36
2.33 HEADERTOKEN ................................................................... 2-37
2.34 HELP ......................................................................................... 2-38
2.35 HISTORY .............................................................................. 2-38
2.36 INTERPRETINTERVAL .......................................................................... 2-38
2.37 INTERPRETTIMESTAMP ............................................................. 2-39
2.38 LOG ........................................................................................ 2-39
2.39 Metadata ddiformat ...................................................................... 2-40
2.40 Metadata defgenformat ............................................................... 2-40
2.41 Metadata detail ........................................................................... 2-40
2.42 Metadata sqlformat...................................................................................................................... 2-41
2.43 NEXT ............................................................................................................................................ 2-41
2.44 NEXTTRAIL .................................................................................................................................. 2-41
2.45 NOTIFY ........................................................................................................................................ 2-41
2.46 OBEY.............................................................................................................................................. 2-42
2.47 OPEN ............................................................................................................................................. 2-43
2.48 POSITION .................................................................................................................................... 2-43
2.49 RECLEN ......................................................................................................................................... 2-44
2.50 SAVE .............................................................................................................................................. 2-44
2.51 SCANFORENDRTRANS .............................................................................................................. 2-46
2.52 SCANFORHEADER ...................................................................................................................... 2-46
2.53 SCANFORMETADATA .................................................................................................................. 2-46
2.54 SCANFORRBA ............................................................................................................................ 2-47
2.55 SCANFORTIME .......................................................................................................................... 2-47
2.56 SCANFORTYPE ........................................................................................................................... 2-48
2.57 SCANSCROLLING ........................................................................................................................ 2-48
2.58 SHOW ........................................................................................................................................... 2-49
2.59 SKIP ................................................................................................................................................ 2-51
2.60 TIME .............................................................................................................................................. 2-51
2.61 TIMEOFFSET .............................................................................................................................. 2-52
2.62 TMFBEFOREIMAGE .................................................................................................................... 2-52
2.63 TMFBEFOREIMAGE .................................................................................................................... 2-53
2.64 TMFGETRECADDR | NOTMFGETRECADDR ............................................................................ 2-53
2.65 TMFIGNORERECCOUNT ............................................................................................................. 2-53
2.66 TRAILFORMAT ........................................................................................................................... 2-54
2.67 TRANSBYTELIMIT ..................................................................................................................... 2-54
2.68 TRANSHIST ................................................................................................................................. 2-54
2.69 TRANSRECLIMIT ......................................................................................................................... 2-55
2.70 USERTOKEN ............................................................................................................................... 2-56
2.71 VOLUME ...................................................................................................................................... 2-56
2.72 WRITELOG ................................................................................................................................... 2-56
2.73 X ...................................................................................................................................................... 2-57
Preface

This guide contains guidelines for using the Oracle GoldenGate Logdump utility. This utility is mainly used by Oracle Support to resolve an Oracle GoldenGate support case. Experienced systems or database administrators who are familiar with the internal structures of Oracle GoldenGate trail records can use Logdump to troubleshoot and resolve errors or other problems encountered during processing.

Audience

This guide is intended for the person or persons who are responsible for operating Oracle GoldenGate and maintaining its performance. This audience typically includes, but is not limited to, systems administrators and database administrators. It also is intended to help Oracle support technicians and other technical representatives when working with Oracle GoldenGate customers.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Information

The Oracle GoldenGate Product Documentation Libraries are found at

Oracle GoldenGate
Oracle GoldenGate Application Adapters
Oracle GoldenGate for Big Data
Oracle GoldenGate Director
Oracle GoldenGate Plug-in for EMCC
Oracle GoldenGate for HP NonStop (Guardian)
Oracle GoldenGate Monitor
Oracle GoldenGate Veridata
Oracle GoldenGate Studio

Additional Oracle GoldenGate information, including best practices, articles, and solutions, is found at:

Oracle GoldenGate A-Team Chronicles

Conventions

The following text conventions are used in this document:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>boldface</strong></td>
<td>Boldface type indicates graphical user interface elements associated with an action, such as &quot;From the File menu, select Save.&quot; Boldface also is used for terms defined in text or in the glossary.</td>
</tr>
<tr>
<td><em>italic</em></td>
<td>Italic type indicates placeholder variables for which you supply particular values, such as in the parameter statement: TABLE table_name. Italic type also is used for book titles and emphasis.</td>
</tr>
<tr>
<td><strong>monospace</strong></td>
<td>Monospace type indicates code components such as user exits and scripts; the names of files and database objects; URL paths; and input and output text that appears on the screen. Uppercase monospace type is generally used to represent the names of Oracle GoldenGate parameters, commands, and user-configurable functions, as well as SQL commands and keywords.</td>
</tr>
<tr>
<td><strong>UPPERCASE</strong></td>
<td>Uppercase in the regular text font indicates the name of a utility unless the name is intended to be a specific case.</td>
</tr>
<tr>
<td>[ ]</td>
<td>Braces within syntax enclose a set of options that are separated by pipe symbols, one of which must be selected, for example: {option1</td>
</tr>
<tr>
<td>[ ]</td>
<td>Brackets within syntax indicate an optional element. For example in this syntax, the SAVE clause is optional: CLEANUP REPLICAT group_name [ , SAVE count]. Multiple options within an optional element are separated by a pipe symbol, for example: [option1</td>
</tr>
</tbody>
</table>
This chapter contains instructions for using the Logdump utility of . Logdump enables you to search for, filter, view, and save data that is stored in a trail or extract file.

**Note:**
To avoid any adverse effects on the data or checkpoints in your trails, use Logdump only with guidance from an Oracle support analyst or an experienced user.

This chapter includes the following sections:
- Getting Started with Logdump
- Viewing the First Record
- Executing Basic Logdump Tasks
- Evaluating Transaction Size
- Maintaining Command History
- Viewing Logdump Command Information and Syntax

### 1.1 Getting Started with Logdump

This section introduces you to basic Logdump commands that enable you to open files, control the display, navigate through a file, and filter for specific information, among other basic tasks. It also illustrates and explains the components of a record.

### 1.2 Viewing the First Record

The steps in this section explain how to set up the Logdump environment and start viewing records.

#### 1.2.1 Running Logdump

Run the `logdump` program from the installation location. Logdump command lines are numbered so that you can use edit and history commands.

#### 1.2.2 Setting Up the View

The following commands set up a Logdump environment that shows the information most commonly used when analyzing trail records.

1. To view the record header with the data:
1.2.3 Opening a Trail File

To open a trail file, complete the following steps:

1. Open a file with the following command:
   Logdump 6> OPEN file_name

   Where:
   
   file_name is either the relative name or fully qualified name of the file, including the file sequence number. For example:
   
   open /home/ggs/dirdat/jd000000
   open $data01.glogggl.aa000000

2. To go to the first record and then move through records in sequence:
   Logdump 7> NEXT

   Alternatively, you can simply enter N.
1.3 Executing Basic Logdump Tasks

The following are some basic tasks that can be performed with Logdump. For detailed information about the commands shown and other available options, see the alphabetical reference in Logdump Commands.

1.3.1 Finding the Next Good Record Header

To find the next good record header, enter the following command:

```
Logdump 8> SCANFORHEADER
```

Alternatively, you can simply enter SFH.

1.3.2 Finding the Beginning, Middle, and End of a Transaction

To find the beginning, middle, and end of a transaction, complete the following steps:
1. Show headers and detail.

   Logdump 9> GHDR ON
   Logdump 10> DETAIL ON

2. Go to the next record.

   Logdump 11> N

3. View the TransInd field in the record header. The following table explains where
   the record is in relation to the transaction.

<table>
<thead>
<tr>
<th>Transaction Indicator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TransInd : . (x00)</td>
<td>First statement in transaction</td>
</tr>
<tr>
<td>TransInd : . (x01)</td>
<td>Statement in middle of transaction</td>
</tr>
<tr>
<td>TransInd : . (x02)</td>
<td>Last statement in transaction</td>
</tr>
<tr>
<td>TransInd : . (x03)</td>
<td>Sole statement in transaction</td>
</tr>
</tbody>
</table>

4. Move through subsequent records by pressing N, and refer to the TransInd field
   to determine where each one is within the transaction. When TransInd is either
   x02 or x03, the TransInd of the next record should be x00, starting a new
   transaction.

1.3.3 Scanning for the End of a Transaction

To scan for the end of a transaction, enter the following command:

Logdump 20> SCANFORENDTRANS

Alternatively, you can simply enter SFET.

The record shown will be the first one in the next transaction. To confirm, the
TransInd field should be x00.

1.3.4 Going to a Specific RBA in the File

To go to a specific RBA in the file, you can do either of the following:

- To go to an RBA anywhere in the file:
  
  Logdump 35> POS rba
  Logdump 36> N

  This displays the record located at that RBA.

- To go to the first record in the file:

  Logdump 37> POS FIRST

  Alternatively, you can enter the following command:
1.3.5 Filtering Based on a Table or Data File Name

To filter out everything except records containing a specific NonStop data file name, enter the following command:

```
Logdump 60> FILTER INCLUDE FILENAME $volume.subvolume.file
```

To filter out everything except records containing a specific table name:

- On a NonStop system:
  
  ```
  Logdump 60> FILTER INCLUDE ANSINAME catalog.schema.table
  ```

- On a Windows or UNIX system:
  
  ```
  Logdump 60> FILTER INCLUDE FILENAME [container | catalog] schema.table
  ```

Now, when you use the N command, you will only see records that satisfy this filter. Conversely, to filter out records containing a specific table or file name, but show everything else, use the EXCLUDE option instead of INCLUDE.

1.3.6 Removing the Current Filter Criteria

To remove the current filter criteria, enter the following command:

```
Logdump 62> FILTER CLEAR
```

1.3.7 Filtering on Multiple Conditions

To filter on multiple conditions, enter one of the following commands:

- Logdump 60> FILTER INCLUDE FILENAME $volume.subvolume...file...; FILTER RECTYPE record_type; FILTER MATCH ALL

- Logdump 60> FILTER INCLUDE ANSINAME catalog.schema.table; FILTER RECTYPE record_type; FILTER MATCH ALL

- Logdump 65> FILTER INCLUDE FILENAME schema.table; FILTER RECTYPE record_type; FILTER MATCH ALL

Use `MATCH ANY` or `MATCH ALL` depending on whether you want the search to match any or all of the filter conditions, respectively, when multiple conditions are specified. The preceding example filters on a name and record type, typically an operation type such as INSERT.

1.3.8 Count the Records in a Trail File

The following command shows a count summary followed by counts for each table or data file:

```
Logdump 67> COUNT
```

1.3.9 Saving Records to a New Trail File

To save a subset of records, enter the following command:

```
Logdump 68> SAVE file n RECORDS
```
1.3.10 Closing the Current File and Opening the Next One in the Trail

To close the current file and open the next one in the trail, enter the following command:

Logdump 69> NEXTTRAIL

1.3.11 Keeping a Log of Your Session

To start logging, enter the following command:

Logdump 70> LOG TO filename.txt

To write text to the log:

Logdump 71> WRITELOG "text"

To stop logging:

Logdump 72> LOG STOP

1.3.12 Seeing the Current Logdump Environment

To see the current Logdump environment, enter the following command:

Logdump 73> ENV

This shows which features are enabled, such as filtering and header views, and it shows environment information such as the current trail and position.

1.3.13 Getting Online Command Help

To get online command help, enter the following command:

Logdump 74> HELP

1.3.14 To Exit Logdump

To exit Logdump, enter either of the following commands:

- Logdump 100> EXIT
- Logdump 100> QUIT

1.4 Evaluating Transaction Size

Use Logdump's TRANSHIST command in conjunction with other Logdump commands to determine whether or not your applications generate large transactions and to identify their relative size. TRANSHIST causes Logdump to track the size of transactions contained in a trail file or extract file in an internal history table. The transactions are ranked in descending order of size, in bytes. When the history table is full, the smallest transaction is removed to allow a larger transaction to be added to the list.

To use statistics generated by TRANSHIST, issue the following series of commands in Logdump:

1. Use TRANSHIST to set the size of the history table that tracks transaction size. The maximum size is 200 bytes. A value of 0 turns off the tracking.
2. Use either the TRANSRECLIMIT or TRANSBYTELIMIT command to set a lower boundary for what is considered a normal sized transaction. These commands prevent normal-sized transactions from being tracked. Eliminating normal-sized transactions reduces the amount of data that must be reviewed.

{TRANSBYTELIMIT n bytes | TRANSRECLIMIT n records}

3. Use Logdump’s COUNT command to display the statistics on transaction size, which appear at the end of the output and look like the following excerpt:

Transactions with at least 100 records or 100000 bytes
2011/02/01 09:31:24.000.000 00:00:00.000, Seq 0, RBA 13101
    Bytes/Trans ...... 1168167
    Records/Trans ... 1001
    Files/Trans ...... 1
2011/02/01 09:31:35.000.000 00:00:11.000, Seq 0, RBA 1205292
    Bytes/Trans ...... 1168167
    Records/Trans ... 1001
    Files/Trans ...... 1

Logdump scans the file(s) and reports the information.

4. Use Logdump’s POSITION RBA command to go to each RBA listed in the COUNT output to find out the name of the table that generated the transaction. You can group these tables into their own processing group so that they do not affect processing of other tables that generate normal sized transactions.

1.5 Maintaining Command History

On Windows and UNIX systems, command history is stored in a file named logdump.hst. On NonStop systems, command history is stored in a file named logduhst. The file is created in the home location of the user who first started Logdump.

When Logdump starts up, it looks for the history file in one of the following locations, depending on the platform:

- The USERPROFILE environment variable.
- The $HOME environment variable.
- The default $vol.subvol.

If the file exists, Logdump loads the command history into a buffer. The command history buffer holds 400 commands. Upon termination of the Logdump session, the session’s history is appended to the file.

1.6 Viewing Logdump Command Information and Syntax

To learn more about Logdump commands, see the alphabetical reference documentation in Logdump Commands.
This chapter contains an alphabetical reference for syntax and usage of the Logdump utility commands.

For Logdump usage instructions, see Using the Logdump Utility.

### 2.1 Logdump Command Summary

The following are category summaries of the Logdump commands.

**Table 2-1 Working with Files**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD</td>
<td>Sets the default directory, volume or subvolume.</td>
</tr>
<tr>
<td>DATAFILE</td>
<td>Opens an Enscribe file from within Logdump.</td>
</tr>
<tr>
<td>GGSAUDITREAD</td>
<td>Reads the TMF audit trail.</td>
</tr>
<tr>
<td>LOG</td>
<td>Writes a session log.</td>
</tr>
<tr>
<td>NEXTTRAIL</td>
<td>Closes the current file and opens the next file in the trail sequence.</td>
</tr>
<tr>
<td>OPEN</td>
<td>Opens a trail file or extract file.</td>
</tr>
<tr>
<td>POSITION</td>
<td>Sets the read position in the file.</td>
</tr>
<tr>
<td>SAVE</td>
<td>Writes record data to another file.</td>
</tr>
<tr>
<td>WRITELOG</td>
<td>Writes text to a session log.</td>
</tr>
<tr>
<td>VOLUME</td>
<td>Sets the default directory, volume or subvolume.</td>
</tr>
</tbody>
</table>

**Table 2-2 Viewing Information**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BULKIOSTATS</td>
<td>Displays bulk I/O statistics.</td>
</tr>
<tr>
<td>COUNT</td>
<td>Displays record count information.</td>
</tr>
<tr>
<td>FILES</td>
<td>Displays file names in the current directory or subvolume.</td>
</tr>
<tr>
<td>ENV</td>
<td>Displays current Logdump settings.</td>
</tr>
<tr>
<td>FILEHEADER</td>
<td>Displays file header information.</td>
</tr>
</tbody>
</table>
### Table 2-2  (Cont.) Viewing Information

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HASHCLEAR</td>
<td>Clears the memory allocated to hash statistics generated when HASHSTATS is enabled.</td>
</tr>
<tr>
<td>HASHSTATS</td>
<td>Enables or disables the display of statistics about file name when a COUNT DETAIL command is issued.</td>
</tr>
<tr>
<td>NOTIFY</td>
<td>Displays the number of records scanned, the trail position, and the record timestamp at specified intervals when using COUNT and records are being suppressed from display through filtering options.</td>
</tr>
<tr>
<td>SHOW</td>
<td>Displays internal information such as the current Logdump environment, a list of record types, and current filter settings.</td>
</tr>
<tr>
<td>TIME</td>
<td>Displays the current time in local and GMT formats.</td>
</tr>
</tbody>
</table>

### Table 2-3  Selecting Data and Records

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEGIN</td>
<td>Searches through a TMF or trail to locate a record at or near the specified time.</td>
</tr>
<tr>
<td>ARGETNONDATACHANGES</td>
<td>ARSTOPNONDATACHANGES</td>
</tr>
<tr>
<td>DUMP</td>
<td>Displays the specified number of bytes of data from the current position in the file.</td>
</tr>
<tr>
<td>FILTER</td>
<td>Filters the display of records.</td>
</tr>
<tr>
<td>NEXT</td>
<td>Displays the next record(s) in the file.</td>
</tr>
<tr>
<td>SCANFORENDTRANS</td>
<td>Finds a record that is the last record of, or the only record in, a transaction, and then displays the first record of the next transaction.</td>
</tr>
<tr>
<td>SCANFORHEADER</td>
<td>Finds the start of the next record header.</td>
</tr>
<tr>
<td>SCANFORMETADATA</td>
<td>Finds a specific metadata record.</td>
</tr>
<tr>
<td>SCANFORRBA</td>
<td>Finds a specific relative byte address.</td>
</tr>
<tr>
<td>SCANFORTIME</td>
<td>Finds the next record with a specific timestamp.</td>
</tr>
<tr>
<td>SCANFORTYPE</td>
<td>Finds the next record of a specific type.</td>
</tr>
<tr>
<td>SKIP</td>
<td>Skips a specified number of records.</td>
</tr>
<tr>
<td>TMFBEFOREIMAGE</td>
<td>Sets the TMFARLIB to fetch the before image of a record and display it with the after image.</td>
</tr>
</tbody>
</table>
### Table 2-4  Caking Conversions

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALCTLFKEY</td>
<td>Calculates a unique key for TLF/PTLF records in ACI's BASE24 application.</td>
</tr>
<tr>
<td>COMPUTETIMESTAMP</td>
<td>Converts a datetime string to a Julian timestamp.</td>
</tr>
<tr>
<td>CTIME</td>
<td>Converts a C timestamp to an ASCII timestamp.</td>
</tr>
<tr>
<td>DECRYPT</td>
<td>Decrypts data before displaying it in Logdump.</td>
</tr>
<tr>
<td>ENCRYPT</td>
<td>Encrypts file data.</td>
</tr>
<tr>
<td>ESBLOCK</td>
<td>Displays NonStop entry-sequenced syskeys as a block number and record number.</td>
</tr>
<tr>
<td>FLOAT</td>
<td>Displays a number or hex string as a hex representation and a floating point number.</td>
</tr>
<tr>
<td>INTERPRETINTERVAL</td>
<td>Displays a 64-bit Julian interval as $dd$-$hh:mm:ss:sss$.</td>
</tr>
<tr>
<td>INTERPRETTIMESTAMP</td>
<td>Displays a 64-bit Julian timestamp in ASCII format.</td>
</tr>
</tbody>
</table>

### Table 2-5  Controlling the Logdump Environment

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCIIDATA</td>
<td>EBCDICDATA</td>
</tr>
<tr>
<td>FMW Generic Topic</td>
<td>Sets the character set on an IBM mainframe.</td>
</tr>
<tr>
<td>FMW Generic Topic</td>
<td>Controls whether the table name is displayed in ASCII or EBCDIC format on an IBM mainframe.</td>
</tr>
<tr>
<td>DETAIL</td>
<td>Controls the display of detailed record information.</td>
</tr>
<tr>
<td>GGSTOKEN</td>
<td>Controls the display of token data.</td>
</tr>
<tr>
<td>GGSTOKEN</td>
<td>Controls the display of automatically generated token data.</td>
</tr>
<tr>
<td>GHDR</td>
<td>Controls the display of header information.</td>
</tr>
<tr>
<td>HEADERTOKEN</td>
<td>Controls the display of header token indicators.</td>
</tr>
<tr>
<td>Metadata ddlformat</td>
<td>Displays the TDR columns in Nonstop DDL format.</td>
</tr>
<tr>
<td>Metadata defgenformat</td>
<td>Displays the columns in a TDR in the format of a GG def from a definition file.</td>
</tr>
<tr>
<td>Metadata detail</td>
<td>Displays tokens and values that make up a metadata record.</td>
</tr>
<tr>
<td>Metadata sqlformat</td>
<td>Displays the columns in a TDR in the format of a SQL table create.</td>
</tr>
</tbody>
</table>
### Table 2-5 (Cont.) Controlling the Logdump Environment

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECLEN</td>
<td>Sets the maximum data output length.</td>
</tr>
<tr>
<td>SCANSCROLLING</td>
<td>Controls whether a count notification displays on one line or multiple lines.</td>
</tr>
<tr>
<td>TIMEOFFSET</td>
<td>Sets the time offset from GMT.</td>
</tr>
<tr>
<td>TMFBEFOREIMAGE</td>
<td>Controls whether or not the before image is displayed for update operations from TMF audit.</td>
</tr>
<tr>
<td>TMFGETRECADDR</td>
<td>NOTMGETRECADDR</td>
</tr>
<tr>
<td>TMFIGNORERECCOUNT</td>
<td>Sets the number of records that the TMFARLIB can ignore before returning a CURRENTPOSITION record.</td>
</tr>
<tr>
<td>TRAILFORMAT</td>
<td>Sets the trail format to the old version (pre-6.0) or the new version.</td>
</tr>
<tr>
<td>TRANSBYTETELIMIT</td>
<td>Sets a byte-count threshold for what is defined as a normal-sized transaction.</td>
</tr>
<tr>
<td>TRANSHIST</td>
<td>Sets the size of the transaction history table that is used for tracking transaction size.</td>
</tr>
<tr>
<td>TRANSRECLIMIT</td>
<td>Sets a record-count threshold for what is defined as a normal-sized transaction.</td>
</tr>
<tr>
<td>USERTOKEN</td>
<td>Controls the display of user token data.</td>
</tr>
</tbody>
</table>

### Table 2-6 Miscellaneous Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEBUG</td>
<td>Turns on Logdump debugging.</td>
</tr>
<tr>
<td>EXIT</td>
<td>Exits Logdump.</td>
</tr>
<tr>
<td>FC</td>
<td>Edits a previous command.</td>
</tr>
<tr>
<td>HELP</td>
<td>Shows syntax for Logdump commands.</td>
</tr>
<tr>
<td>HISTORY</td>
<td>Lists previously issued commands.</td>
</tr>
<tr>
<td>OBEY</td>
<td>Executes a series of commands stored in a file.</td>
</tr>
<tr>
<td>X</td>
<td>Executes a program from within Logdump.</td>
</tr>
</tbody>
</table>

#### 2.2 ARGETNONDATACHANGES | ARSTOPNONDATACHANGES

Use ARGETNONDATACHANGES and ARSTOPNONDATACHANGES when reading TMF trails to direct TMFARLIB to include or exclude records that did not result in a change to the database.
ARGETNONDATACHANGES

Syntax
ARGETNONDATACHANGES | ARSTOPNONDATACHANGES

ARGETNONDATACHANGES
Includes non-change records.

ARSTOPNONDATACHANGES
Suppresses non-change records.

2.3 ASCIIDATA | EBCDICDATA

Use ASCIIDATA or EBCDICDATA to control whether record data is displayed in ASCII or EBCDIC format on an IBM mainframe. You may need to use the ASCIIDUMP command to set the character set first.

Default
ASCIIDATA ON (same as EBCDICDATA OFF)

Syntax
ASCIIDATA {ON | OFF} | EBCDICDATA {ON | OFF}

2.4 FMW Generic Topic

This is the start of your topic.

2.5 FMW Generic Topic

This is the start of your topic.

2.6 BEGIN

Use BEGIN to do a binary search through a TMF or trail to locate a record at or near the specified time.

Default
None

Syntax
BEGIN timestamp

timestamp
The timestamp for which to search.
2.7 BULKIOSTATS

Use BULKIOSTATS to display statistics for bulk I/O records, including the number of physical and logical reads and current and average read times.

Default
None

Syntax
BULKIOSTATS

2.8 CALCTLFKEY

Use CALCTLFKEY to work around the @GETENV ("TLFKEY", SYSKEY "unique_key") function when it generates duplicate keys. The TLFKEY option associates a unique key with TLF/PTLF records in ACI's BASE24 application. CALCTLFKEY performs the calculation and displays the value for each record.

Without an argument, CALCTLFKEY displays whether this feature is enabled or disabled.

Default
OFF

Syntax
CALCTLFKEY {ON | OFF}

2.9 CD

Use CD to set the default directory, volume, or subvolume. An alias for this command is VOLUME.

Default
None

Syntax
CD {directory | volume | subvolume}

2.10 COMPUTETIMESTAMP

Use COMPUTETIMESTAMP to convert a datetime string to Julian format.

Default
None

Syntax
COMPUTETIMESTAMP string
string
A datetime string in the format of:

[[yy]yy-mm-dd] [hh:mm:ss]]

Example

COMPUTETIMESTAMP 2005-01-01 12:00:00

This example returns the following:

2005-01-01 12:00:00 is JulianTimestamp 211971340800000000

2.11 COUNT

Use COUNT to produce a record count summary and other information related to the amount of data in the file. The basic output, without options, shows the following:

- The RBA where the count began
- The total data bytes and average bytes per record
- Information about the operation types
- Information about the transactions

When the DETAIL command is issued prior to issuing COUNT, the information includes a count for each table or data file. COUNT options allow you to show table detail without using the DETAIL command first, set a start and end time for the count, filter the count for a table, data file, trail file, or extract file, and specify a time interval for counts. You can use the DETAIL OFF command to turn off the collection of this detailed information.

For arguments that take a time string, use the following format:

[[yy]yy-mm-dd] [hh:mm:ss]]

Default

Produce a count summary of all records.

Syntax

COUNT
{, DETAIL}
{, END[TIME] time_string]
{, INT[ERVAL] minutes]
{, LOG] wildcard]
{, START[TIME] time_string]

DETAIL

Adds a count for each table or data file that was processed by Extract to the summary count. The information includes the total and average number of data bytes and information about the operations that were performed. This data can also be obtained by using the DETAIL command before issuing COUNT.

END[TIME] time_string

Stops the count with the last record written at the specified time.
**INT[ERVAL]** minutes
Displays statistics for total bytes, average bytes, and number of each type of operation that occurred within a specified interval of time, in minutes. Then it displays the totals for those statistics.

**LOG wildcard**
Produces a count for multiple trail or extract files specified with a wildcard.

**START[T]IME** time_string
Begins the count with the first record written at the specified time.

**Examples**

**Example 1**
COUNT START 2011-01-11 12:00:00 , END 2011-01-12 12:00:00

**Example 2**
COUNT INTERVAL 4

This displays something similar to the following. Individual table or data file count has been truncated due to space constraints.

On Windows or UNIX (file names will differ):

Interval from 2011/02/28 11:30:00.000 to 2011/02/28 11:34:00.000,
Recs 3
Total Data Bytes 120
Avg Bytes/Record 40
Delete 3
Before Images 3
LogTrail /home/ggs/dirdat/rt000000 has 304 records
Total Data Bytes 12120
Avg Bytes/Record 39
Delete 3
Insert 300
FieldComp 1
Before Images 3
After Images 301

Average of 303 Transactions
Bytes/Trans ...... 88
Records/Trans ... 1
Files/Trans ...... 1

HR.JOBS Partition 4
Total Data Bytes 5911
Avg Bytes/Record 68
Insert 86
After Images 86

HR.REGIONS Partition 4
Total Data Bytes 512
Avg Bytes/Record 32
Insert 16
After Images 16

On NonStop:
LogTrail \GGQA.$QA01.QADAT.LS000000 has 29656 records
Total Data Bytes 3561022
    Avg Bytes/Record 120
Delete 50
    Insert 21221
Update 8379
GSSPurgedata 6
Before Images 50
After Images 29606

Average of 3621 Transactions
    Bytes/Trans ...... 1376
    Records/Trans ... 8
    Files/Trans ...... 1

\GGQA.$QA01.QAESRC.ACCTS                          Partition 0
Total Data Bytes 286414
    Avg Bytes/Record 142
Delete 17
    Insert 2000
Before Images 17
After Images 2000

\GGQA.$QA01.QAESRC.ACCTN                          Partition 0
Total Data Bytes 281700
    Avg Bytes/Record 100
Delete 17
    Insert 2000
Update 800
Before Images 17
After Images 2800

Example 3

COUNT LOG ls*

This produces a count for all files whose names begin with LS. (Individual table or data file count has been truncated due to space constraints.)

On NonStop:

Current LogTrail is \GGQA.$QA01.QADAT.LS000000
Bad record found at RBA 5287, format 5.50)
2A56 623F | *Vb?
LogTrail \GGQA.$QA01.QADAT.LS000000 has 33 records
LogTrail \GGQA.$QA01.QADAT.LS000000 closed
Current LogTrail is \GGQA.$QA01.QADAT.LS000001
LogTrail \GGQA.$QA01.QADAT.LS000001 has 99 records
LogTrail \GGQA.$QA01.QADAT.LS000001 closed
Current LogTrail is \GGQA.$QA01.QADAT.LS000002
LogTrail \GGQA.$QA01.QADAT.LS000002 has 0 records
LogTrail \GGQA.$QA01.QADAT.LS000002 closed
Current LogTrail is \GGQA.$QA01.QADAT.LS000003
LogTrail \GGQA.$QA01.QADAT.LS000003 has 0 records
LogTrail \GGQA.$QA01.QADAT.LS000003 closed
LogTrail \GGQA.$QA01.QADAT.LS* has 132 records

Total Data Bytes 9468
    Avg Bytes/Record 71
    Insert 132
After Images 132

Average of 4 Transactions
Bytes/Trans ..... 3951
Records/Trans ... 33
Files/Trans ..... 3

QAHRTS.JOBS
Total Data Bytes 5220
Avg Bytes/Record 68
Insert 76
After Images 76

On Windows or UNIX (file names will differ):
Current LogTrail is c:\goldengate802\dirdat\ls000000
Bad record found at RBA 5287, format 5.50)
   2A56 623F | *Vb?
LogTrail c:\goldengate802\dirdat\ls000000 has 33 records
LogTrail c:\goldengate802\dirdat\ls000000 closed
Current LogTrail is c:\goldengate802\dirdat\ls000001
LogTrail c:\goldengate802\dirdat\ls000001 has 99 records
LogTrail c:\goldengate802\dirdat\ls000001 closed
Current LogTrail is c:\goldengate802\dirdat\ls000002
LogTrail c:\goldengate802\dirdat\ls000002 has 0 records
LogTrail c:\goldengate802\dirdat\ls000002 closed
Current LogTrail is c:\goldengate802\dirdat\ls000003
LogTrail c:\goldengate802\dirdat\ls000003 has 0 records
LogTrail c:\goldengate802\dirdat\ls000003 closed
LogTrail c:\goldengate802\dirdat\ls* has 132 records

Total Data Bytes 9468
Avg Bytes/Record 71
Insert 132
After Images 132

Average of 4 Transactions
Bytes/Trans ..... 3951
Records/Trans ... 33
Files/Trans ..... 3

HR.JOBS
Total Data Bytes 5220
Avg Bytes/Record 68
Insert 76
After Images 76

Example 4

COUNT DETAIL

This produces something similar to the following:
On Windows or UNIX (file names will differ):
LogTrail /home/ggs/dirdat/rt000000 has 304 records
Total Data Bytes 12120
Avg Bytes/Record 39
Delete 3
Insert 300  
FieldComp 1  
Before Images 3  
After Images 301  

Average of 303 Transactions  
  Bytes/Trans ...... 88  
  Records/Trans ... 1  
  Files/Trans ...... 1  

GGS.TCUSTMER  
  Total Data Bytes 12120  
  Avg Bytes/Record 39  
  Delete 3  
  Insert 300  
  FieldComp 1  
  Before Images 3  
  After Images 301  

Files 1, Coll 0, Chain 0  

On NonStop:  
LogTrail \\GGQA.\$QA01.QADAT.LS000000 has 29656 records  
Total Data Bytes 3561022  
  Avg Bytes/Record 120  
  Delete 50  
  Insert 21221  
  Update 8379  
  GSSPurgedata 6  
  Before Images 50  
  After Images 29606  

Average of 3621 Transactions  
  Bytes/Trans ...... 1376  
  Records/Trans ... 8  
  Files/Trans ...... 1  

\\GGQA.\$QA01.QAESRC.ACCTS  
  Total Data Bytes 286414  
  Avg Bytes/Record 142  
  Delete 17  
  Insert 2000  
  Before Images 17  
  After Images 2000  

2.12 CTIME  

Use CTIME to convert a C timestamp to an ASCII timestamp.  

Default  
None  

Syntax  
CTIME C_timestamp_string
2.13 DATAFILE

Use DATAFILE to open an Enscribe data file from within Logdump.

**Note:**
You can use the WRITEREC command to write a record to the Enscribe file.

**Default**
None

**Syntax**
DATAFILE file

**Example**
DATAFILE \SYSA.$DATA04.SALES.CUSTMER

2.14 DEBUG

Use DEBUG to run debugging for Logdump. Use this command with the guidance of a support analyst.

**Default**
Disabled

**Syntax**
DEBUG

2.15 DECRYPT

Use DECRYPT to decrypt data that was encrypted with trail encryption, so that it can be viewed with Logdump.

**Default**
OFF

**Syntax**
DECRIPT [OFF | ON [KEYNAME key_name]]
**OFF**
No decryption. This is the default.

**ON [KEYNAME key_name]**

- **ON** without the **KEYNAME** clause decrypts data that was encrypted with the **ENCRYPTTRAIL** parameter without options (256-key byte substitution).

- **ON** with the **KEYNAME** clause decrypts data that was encrypted with **ENCRYPTTRAIL** using an AES algorithm and a **KEYNAME** clause. For **key_name**, supply the logical name of the encryption key that was used in the **KEYNAME keyname** clause of **ENCRYPTTRAIL**.

  For more information about **ENCRYPTTRAIL**, see *Reference for Oracle GoldenGate for Windows and UNIX*.

- **ON** also works to decrypt records that were encrypted using the Oracle wallet and master key methods; similar to how the **DECRIPTTRAIL** parameter for Replicat or Pump behaves.

**Example**

DECRYPT ON

**Example**

DECRYPT ON KEYNAME mykey

### 2.16 DETAIL

Use **DETAIL** to include additional information in the Logdump output. By default, Logdump only shows the hex and ASCII representation of the record.

Without options, **DETAIL** displays the status of record detail (**ON** or **OFF**). Options do the following:

- **DETAIL ON** displays a list of columns that includes the column ID, length, and value in hex and ASCII.

- **DATA** adds hex and ASCII data values to the column list.

- **DETAIL OFF** turns off detailed display; **OFF** is the default.

  **DETAIL** can be shortened to **DET**.

**Default**

Display a column list

**Syntax**

```
DETAIL {ON | OFF | DATA}
```

**ON**

Shows detailed column information.

**OFF**

Suppresses detailed column information.
**DATA**
Adds the hex and ASCII data values to the column information.

### 2.17 DUMP

Use **DUMP** to display a HEX/ASCII or HEX/EBCDIC dump of the specified number of bytes from the open trail or extract file, starting at the current RBA.

**DUMP** does not work when reading TMF audit trails, because I/O to the TMF trails is done by TMFARLIB.

**Default**
256

**Syntax**

```
DUMP bytes
```

**bytes**
The number of bytes forward to display. Valid values are from 1 through 28672.

**Example**

```
DUMP 300
```

This produces something similar to the following example. Note: This example shows only a few lines of the record.

```
Dump  300  Bytes at RBA 0
4700 0047 4800 003B 4500 0041 0000 646D 02F1 3387 | G..GH..;E..A..dm..3.
841D FE98 0000 0000 0000 0000 5EA8 DC3C 0352 0000 | ...........^..<.R..
0000 5C54 5249 4L4C 2E24 5141 3031 2E5A 4C5C 2E41 | ..\TRILL.$QA01.QASRC
2E41 4343 544E 005A 0000 4747 0000 4748 0000 3B45 | .ACCTN..GG..GH..;E
```

### 2.18 ENCRYPT

Use **ENCRYPT** to encrypt text supplied as an argument. The encryption method is 256-key byte substitution. The results are printed to screen.

**Default**
None

**Syntax**

```
ENCRYPT text
```

**Example**

```
ENCRYPT 123456789
```

This produces the following:

```
Before
3132 3334 3536 3738 39 | 123456789
```
2.19 ENV

Use ENV to show current Logdump settings.

Default

None

Syntax

ENV

Example

The following shows typical ENV settings on Windows and on NonStop.

Windows (UNIX is similar except for file names):

Current Volume : C:\GoldenGate 002
LogTrail : C:\goldengate002\dirdat\jd000001
Trail Format : New
End of File : 15861
Current Position : 0
Next Position : 0
Last Modtime : 2013/01/01 13:45:51.000.000
Display RecLen : 140
Logtrail Filter : On
Show Ghdr : On
Detail : On
Trans History : 0 Transactions, Records 100, Bytes 100000
LargeBlock I/O : On, Blocksize 57344
Local System : Little Endian
Logtrail Data : Big Endian/ASCII
Logtrail Headers : ASCII
Dump : ASCII
Timeoffset : LOCAL
Scan Notify Interval : 10000 records, Scrolling On

NonStop:

Current Volume : $QU01.QAGGS
LogTrail : \TRGGS.$QA01.QADAT.LS000000
Trail Format : New
End of File : 5831722
Current Position : 0
Next Position : 0
Last Modtime : 2013/01/01 12:47:11.686.219
Display RecLen : 140
Logtrail Filter : On
Detail : On
Trans History : 0 Transactions, Records 100, Bytes 100000
LargeBlock I/O : On, Blocksize 57344
Local System : Big Endian
Logtrail Data : Big Endian/ASCII
Logtrail Headers : ASCII
Dump : ASCII
Timeoffset : LOCAL
Scan Notify Interval : 10000 records, Scrolling On
2.20 ESBLOCK

Use ESBLOCK for debugging on a NonStop system. It displays an entry-sequenced syskey as a block number and record number.

Default
None

Syntax
ESBLOCK entry_sequenced_RBA

Example
ESBLOCK 4294967302

This produces the following. It shows that the syskey value 4294967302 evaluates to block 1, record 6.

<table>
<thead>
<tr>
<th>Interpretation</th>
<th>Value</th>
<th>Hexadecimal</th>
<th>Block</th>
<th>Record</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interpreted</td>
<td>4294967302</td>
<td>0x00000001000000060</td>
<td>1.6</td>
<td></td>
</tr>
<tr>
<td>64-bit Syskey</td>
<td>4294967302</td>
<td>0x00000001000000060</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ES64_TO_RBA64</td>
<td>4102</td>
<td>0x00000000000010060</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RBA64_TO_ES64</td>
<td>4294967302</td>
<td>0x00000001000000060</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2.21 EXIT

Use EXIT to exit Logdump and terminate the process. An alias for EXIT is QUIT.

Default
None

Syntax
EXIT

2.22 FC

Use FC to edit a previously issued Logdump command and then execute it again. Previous commands are stored in the memory buffer and can be displayed by issuing the HISTORY command (see “HISTORY”). Issuing FC without arguments executes the most recently used command. By using options, you can retrieve a specific command by specifying its line number or a text substring.

Using the Editor

The FC command displays the command and then opens an editor with a prompt containing a blank line starting with two dots. To edit a command, use the space bar to position the cursor beneath the character where you want the change to begin, and then enter one of the following arguments. Arguments are not case-sensitive and can be combined.
### Table 2-7 FC Command Options

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>i</strong></td>
<td>Inserts text. For example:</td>
</tr>
<tr>
<td>Logdump 24&gt; fc 9</td>
<td>&gt; count</td>
</tr>
<tr>
<td>..</td>
<td>&gt; i detail</td>
</tr>
<tr>
<td>count detail</td>
<td></td>
</tr>
<tr>
<td><strong>r</strong></td>
<td>Replaces text. For example:</td>
</tr>
<tr>
<td>Logdump 25&gt; fc 10</td>
<td>&gt; timeoffset local</td>
</tr>
<tr>
<td>..</td>
<td>&gt; rgmt</td>
</tr>
<tr>
<td>timeoffset gmt</td>
<td></td>
</tr>
<tr>
<td><strong>d</strong></td>
<td>Deletes a character. To delete multiple characters, enter a d for each one. For example:</td>
</tr>
<tr>
<td>Logdump 26&gt; fc 11</td>
<td>&gt; scanforrrbba</td>
</tr>
<tr>
<td>..</td>
<td>&gt; dd</td>
</tr>
<tr>
<td>scanforrbba</td>
<td></td>
</tr>
<tr>
<td>replacement_text</td>
<td>Replaces the displayed command with the text that you enter on a one-for-one basis. For example:</td>
</tr>
<tr>
<td>Logdump 26&gt; fc 10</td>
<td>&gt; scanforrrbba 107</td>
</tr>
<tr>
<td>..</td>
<td>&gt; 127</td>
</tr>
<tr>
<td>scanforrbba 127</td>
<td></td>
</tr>
</tbody>
</table>

To execute the command, press **Enter** twice, once to exit the editor and once to issue the command. To cancel an edit, type a forward slash (/) twice.

**Default**
Execute the most recent command again

**Syntax**

FC [n | -n | string]

**n**
Returns the specified command line. Each Logdump command line is sequenced, beginning with 1 at the start of the session.

**-n**
Returns the command that was issued n lines before the current line.

**string**
Returns the last command that starts with the specified text string.
Examples

Example 1
FC 9

Example 2
FC -3

Example 3
FC sca

2.23 FILEHEADER

Use FILEHEADER to display the contents of the header of the currently open trail file. The file header is stored as a record at the beginning of a trail file preceding the data records. The information that is stored in the trail header provides enough information about the records to enable an Oracle GoldenGate process to determine whether the records are in a format that the current version of Oracle GoldenGate supports.

The trail header fields are stored as tokens, where the token format remains the same across all versions of Oracle GoldenGate. If a version of Oracle GoldenGate does not support any given token, that token is ignored. Deprecated tokens are assigned a default value to preserve compatibility with previous versions of Oracle GoldenGate.

The current FILEHEADER command applies globally to the Logdump session, until a different FILEHEADER command is issued.

To View the File Header

To view the file header:

1. Position to the beginning of the trail file with the following Logdump command.
   pos 0

2. Issue the following Logdump command to see the first record of the file, the one that contains the file header.
   next

To Retrieve the File Header Tokens

To retrieve file header values as input parameters, use the @GETENV function with the GGFILEHEADER option. See Reference for Oracle GoldenGate for Windows and UNIX.

Note:
The Logdump command HEADERTOKEN also shows trail tokens, but it shows a brief summary of each one. FILEHEADER shows actual token values.
### Table 2-8  File Header Tokens

<table>
<thead>
<tr>
<th>Token/subtoken</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signature</td>
<td>UINT32</td>
<td>Internal use.</td>
</tr>
<tr>
<td>Compatibility</td>
<td>UINT16</td>
<td>The version of the trail. The compatibility level of the software must be greater than, or equal to, that of the trail file for a process to be able to read the trail file. Current valid values are 0 to 5.</td>
</tr>
<tr>
<td>CharSet</td>
<td>INT32</td>
<td>The global character set of the trail file, as defined in the parameter file or the default value. For example: WCP1252-1 -3 indicates the system default.</td>
</tr>
<tr>
<td>CreationTime</td>
<td>Timestamp</td>
<td>The time that the trail file was created, in local GMT Julian time, INT 64.</td>
</tr>
</tbody>
</table>
| URI            | String    | The universal resource identifier of the process that created the trail file, in the format of:  

`host_name:dir[:dir][:dir]group_name`

Where:
- `host_name` is the name of the server that hosts the process
- `dir` is a subdirectory of the installation path.
- `group_name` is the name of the process group that is linked with the process.

Example:

`syl:home:oracle:v9.5:extora`

Shows where the trail was processed and by which process. This includes a history of previous runs. |
| URIHistory     | String array | List of the URIs of processes that wrote to the trail file before the current process.  

- For a primary Extract, this field is empty.  
- For a data pump, this field is URIHistory + URI of the input trail file. |
<p>| FileName       | String     | Name of the trail file. Can be absolute or relative path, with forward or backward slash depending on the file system. |
| MultiPart      | Boolean    | True/False flag indicating whether the trail file is a single file (such as one created for a batch run) or a sequentially numbered file that is part of a trail for online, continuous processing. If False, the SeqNum subtoken is not valid. |
| SeqNum         | UINT32     | The sequence number of the file in the trail, if MultiPart is true. Invalid if multipart is false. The value is the numerical sequence number, without any zero padding. |
| FileSize       | UINT64     | Size of the trail file. Value is NULL until the trail file is completed. Non-NULL values are in bytes. |</p>
<table>
<thead>
<tr>
<th>Token/subtoken</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FirstRecordCSN</td>
<td>CSN</td>
<td>The commit sequence number (CSN) of the first record in the trail file. Value is NULL until the trail file is completed.</td>
</tr>
<tr>
<td>LastRecordCSN</td>
<td>CSN</td>
<td>The commit sequence number (CSN) of the last record in the trail file. Value is NULL until the trail file is completed.</td>
</tr>
<tr>
<td>FirstRecordIOTime</td>
<td>Timestamp</td>
<td>The time that the first record in the trail file was written. Value is NULL until the trail file is completed.</td>
</tr>
<tr>
<td>LastRecordIOTime</td>
<td>Timestamp</td>
<td>The time that the last record in the trail file was written. Value is NULL until the trail file is completed.</td>
</tr>
<tr>
<td>SysName</td>
<td>String</td>
<td>The name of the operating system, for example: SunOS, Linux, Microsoft Windows</td>
</tr>
<tr>
<td>NodeName</td>
<td>String</td>
<td>The name of the machine, for example sys1.</td>
</tr>
<tr>
<td>Release</td>
<td>String</td>
<td>The release level of the operating system, for example: 5.10, 2.6.9-11.ELsmp</td>
</tr>
<tr>
<td>Version</td>
<td>String</td>
<td>The version of the operating system, for example: s10_69, #1 SMP Fri Feb 24 16:56:28 EST 2006</td>
</tr>
<tr>
<td>Hardware</td>
<td>String</td>
<td>The hardware type of the processor, for example: sun4u, x86_64, x86</td>
</tr>
<tr>
<td>Vendor</td>
<td>UINT16</td>
<td>The name of the database vendor. Some (but not all) examples are: DB2 ZOS, CTREE, MSSQL, MYSQL, ORACLE, SQLMX, SYBASE, TERADATA, TIMES TEN, NONSTOP, ENSCRIBE</td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>The name of the database, for example findb.</td>
</tr>
<tr>
<td>Instance</td>
<td>String</td>
<td>The name of the database instance, if applicable to the database type, for example ORA1022A.</td>
</tr>
<tr>
<td>Charset</td>
<td>INT32</td>
<td>The character set of the database. Currently, the valid value is -1 (unknown). (For some databases, this will be empty.)</td>
</tr>
<tr>
<td>Token/subtoken</td>
<td>Data Type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>MajorVersion</td>
<td>UINT16</td>
<td>The major version of the database.</td>
</tr>
<tr>
<td>MinorVersion</td>
<td>UINT16</td>
<td>The minor version of the database.</td>
</tr>
<tr>
<td>VerString</td>
<td>String</td>
<td>The maintenance (patch) level of the database.</td>
</tr>
<tr>
<td>ClientCharset</td>
<td>INT32</td>
<td>The character set of the database client. Currently, the valid value is -1 (unknown). (For some databases, this will be empty.)</td>
</tr>
<tr>
<td>ClientVerString</td>
<td>String</td>
<td>The maintenance (patch) level of the database client. (For some databases, this will be empty.)</td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>The group name that is associated with the process.</td>
</tr>
<tr>
<td>DataSource</td>
<td>UINT16</td>
<td>The data source that was read by the process. Can be one of:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DS_EXTRACT_TRAILS (source was an extract file, populated with change data)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DS_LOG_TABLE (source was a log table, used for trigger-based extraction)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DS_DATABASE (source was a direct select from database table written to a trail, used for SOURCEISTABLE-driven initial load)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DS_TRAN_LOGS (source was the database transaction log)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DS_INITIAL_DATA_LOAD (source was Extract; data taken directly from source tables)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DS_VAM_EXTRACT (source was a vendor access module)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DS_VAM_TWO_PHASE_COMMIT (source was a VAM trail)</td>
</tr>
<tr>
<td>MajorVersion</td>
<td>UINT16</td>
<td>The major version of the process (xx).</td>
</tr>
<tr>
<td>MinorVersion</td>
<td>UINT16</td>
<td>The minor version of the process (xx.xx).</td>
</tr>
<tr>
<td>MaintenanceLevel</td>
<td>UINT16</td>
<td>The maintenance version of the process (xx.xx.xx).</td>
</tr>
<tr>
<td>PatchLevel</td>
<td>UINT16</td>
<td>The patch version of the process (xx.xx.xx.xx).</td>
</tr>
<tr>
<td>BuildNumber</td>
<td>UINT16</td>
<td>The build number of the process.</td>
</tr>
<tr>
<td>VerString</td>
<td>String</td>
<td>The version string of the process. For example: 11.1.1.17A not for production</td>
</tr>
<tr>
<td>RecoveryMode</td>
<td>UINT16</td>
<td>Internal use</td>
</tr>
<tr>
<td>LastCompletedCSN</td>
<td>CSN</td>
<td>Internal use</td>
</tr>
<tr>
<td>LastCompletedXids</td>
<td>Xid</td>
<td>Internal use</td>
</tr>
<tr>
<td>LastSCN</td>
<td>CSN</td>
<td>Internal use</td>
</tr>
<tr>
<td>LastXid</td>
<td>Xid</td>
<td>Internal use</td>
</tr>
</tbody>
</table>

**Default**

OFF
Syntax

FILEHEADER \{ON | OFF | DETAIL\}

ON
Enables the display of the file header, showing the main header tokens.

OFF
Disables the display of the file header.

DETAIL
Provides detailed information that includes the sub-tokens.

Examples

Example 1

FILEHEADER ON
TokenID x46 'F' Record Header    Info x00  Length  587
TokenID x30 '0' TrailInfo        Info x00  Length  303
TokenID x31 '1' MachineInfo      Info x00  Length  103
TokenID x32 '2' DatabaseInfo     Info x00  Length   88
TokenID x33 '3' ProducerInfo     Info x00  Length   85
TokenID x34 '4' ContinuityInfo   Info x00  Length    4
TokenID x5a 'Z' Record Trailer   Info x00  Length  587
2011/1/18 13:39:18.951.346 FileHeader           Len   587 RBA 0
Name: *FileHeader*

GroupID x30 '0' TrailInfo        Info x00  Length  303
3000 012f 3000 0008 660d 0a71 3100 0006 0001 3200 | 0../0...f..q1.....2.
0008 0000 0016 3300 000c 02f1 7834 eac7 7f3f 3400 | .........x4......4.
0037 0031 7572 693a 7465 6c6c 7572 6961 6e3a 3a68 | .7.uri:tellurian:h
6f6d 653a 6d63 6361 7267 6172 3a67 6733 3a67 6773 | ome:mccargar:ggs:ggs
4f72 6163 6c65 3a73 6f75 7263 6536 0000 1700 112e | Oracle:source6.....
2f64 6972 6461 742f 6572 3030 3030 3030 3700 0005 | /dirdat/er0000007...
0138 0000 0800 01e2 4039 0000 0c00 0000 0000 01d | .8......@9..........

GroupID x31 '1' MachineInfo      Info x00  Length  103
3100 0067 3000 000b 0005 4c69 6e75 7831 0000 0700 | 1..g0.....Linux1...
0974 656c 6c75 7269 616e 3200 0014 000e 322e 362e | .tellurian2.....2.6.

...
Logdump Commands

Example 2

FILEHEADER DETAIL

TokenID x46 'F' Record Header
TokenID x30 'O' TrailInfo
TokenID x31 'L' MachineInfo
TokenID x32 '2' DatabaseInfo
TokenID x33 '3' ProducerInfo
TokenID x34 '4' ContinuityInfo
TokenID x5a 'Z' Record Trailer

2011/01/18 13:40:26.034.631 FileHeader Len 587 RBA 0

Name: *FileHeader*

3000 012f 3000 0008 660d 0a71 3100 0006 0001 3200 | 0../0...f..q1.....2.
0008 0000 0016 3300 000c 02f1 7834 7f3f 3400 | .........3.....x4...?
0037 0031 7572 693a 7465 6c6c 7572 6961 6e3a 3a68 | .7.uri:tellurian::h
6f6d 653a 6d63 6361 7267 6172 3a67 6773 3a67 6773 | ome:mccargar:ggs:ggs
4f72 6163 6c65 3a73 6f75 7263 6536 0000 1700 112e | oracle:source6......
2f64 6972 6461 742f 6572 3030 3030 3030 3700 0005 | /dirdat/er0000007...
0138 0000 0000 01e2 4039 0000 0c00 001d .8.....09.....

GroupID x30 'O' TrailInfo

3000 012f 3000 0008 660d 0a71 3100 0006 0001 3200 | 0../0...f..q1.....2.
0008 0000 0016 3300 000c 02f1 7834 7f3f 3400 | .........3.....x4...?
0037 0031 7572 693a 7465 6c6c 7572 6961 6e3a 3a68 | .7.uri:tellurian::h
6f6d 653a 6d63 6361 7267 6172 3a67 6773 3a67 6773 | ome:mccargar:ggs:ggs
4f72 6163 6c65 3a73 6f75 7263 6536 0000 1700 112e | oracle:source6......
2f64 6972 6461 742f 6572 3030 3030 3030 3700 0005 | /dirdat/er0000007...
0138 0000 0000 01e2 4039 0000 0c00 001d .8.....09.....

GroupID x5a 'Z' Record Trailer
2.24 FILES

Use FILES to display summary file information for files on the local system. The default command displays all files in the current directory or subvolume. To constrain the display to specific files, you can supply a wildcarded name.

This command can be shortened to FI. An alias for this command is DIR or FILEINFO.

Default

Show all files in current directory or subvolume
Syntax

FILES [directory | subvolume | volume.subvolume]

directory | subvolume | volume.subvolume
The name of a directory or subvolume or a wildcard for specific files.
If any file or directory in the specified path contains spaces, the entire path must be enclosed within double quotation marks.

Example

FILES "c:\goldengate ver802\dirdat\cc**"

Example

FILES $QA01.*

2.25 FILTER

Use FILTER to filter the display based on one or more criteria.

- You can string multiple FILTER commands together, separating each one with a semi-colon, as in:

  FILTER INCLUDE FILENAME fin.act*; FILTER RECTYPE 5; FILTER MATCH ALL

  Or...

  FILTER INCLUDE FILENAME $QA01.QAESRC.ACCTN; FILTER SYSKEY 4294967302; FILTER MATCH ALL

- To avoid unexpected results, avoid stringing filter options together with one FILTER command. For example, the following would be incorrect:

  FILTER INCLUDE FILENAME fin.act*; RECTYPE 5; MATCH ALL

  Or...

  FILTER INCLUDE FILENAME $QA01.QAESRC.ACCTN; SYSKEY 4294967302

Without arguments, FILTER displays the current filter status (ON or OFF) and any filter criteria that are in effect.

Comparison Operators

For options that take comparison operators, the following standard operators may be used. The absence of an operator implies Equal.

Table 2-9 Filter Option Comparison Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equal</td>
<td>=</td>
</tr>
<tr>
<td></td>
<td>EQ</td>
</tr>
<tr>
<td></td>
<td>==</td>
</tr>
</tbody>
</table>
### Table 2-9 (Cont.) Filter Option Comparison Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Less than</td>
<td>&lt;</td>
</tr>
<tr>
<td></td>
<td>LT</td>
</tr>
<tr>
<td>Less than or equal</td>
<td>&lt;=</td>
</tr>
<tr>
<td></td>
<td>LE</td>
</tr>
<tr>
<td>Greater than</td>
<td>&gt;</td>
</tr>
<tr>
<td></td>
<td>GT</td>
</tr>
<tr>
<td>Greater than or equal</td>
<td>&gt;=</td>
</tr>
<tr>
<td></td>
<td>GE</td>
</tr>
<tr>
<td>Not equal</td>
<td>&lt;&gt;</td>
</tr>
<tr>
<td></td>
<td>NE</td>
</tr>
<tr>
<td></td>
<td>!=</td>
</tr>
</tbody>
</table>

#### Default

Shows current filter settings

#### Syntax

FILTER [INCLUDE] [EXCLUDE] filter_option

Where:

`filter_option` can be one of:

- ANSINAME name [, name] |
- AUDITRBA rba [comparison_operator] |
- CLEAR {filter_spec | ALL} |
- CSN | LogCSN [comparison_operator] {value} |
- ENDTIME time_string |
- FILENAME name [, name] |
- GGSTOKEN token_name [comparison_operator] {token_value} |
- HEX "hex_string" [byte_range] | "hex_string" [byte_range] | [...]
- INT16 16_bit_integer |
- INT32 32_bit_integer |
- IOTYPE operation_type [, operation_type] |
- MATCH {ANY | ALL} |
- OFF |
- ON |
- PROCESS process_name |
- RBA byte_address [comparison_operator] {...} |
- RECLEN length [comparison_operator] |
- RECTYPE {type_number | type_name}
SHOW | 
STARTTIME time_string | 
STRING [BOTH] [B],text [column_range ] 
[[B],text [column_range]] [... ] | 
SYSKEY system_key [comparison_operator] [...] | 
TRANSID transaction_identifier | 
TRANSIND indicator [comparison_operator] | 
TYPE type | 
UNDOFLAG type [comparison_operator] | 
USERTOKEN token_name [comparison_operator] [token_value] }

**ANSINAME name [ , name]**
Filters based on the ANSI name of a SQL/MX table or a table from a Windows or UNIX source system. For use on NonStop systems. The format for *name* is:
catalog.schema.table
Up to eight name specifications may be supplied. The command is case-sensitive.
To filter based on the name of a data file, use the FILENAME option.

**AUDITRBA rba [comparison_operator]**
Filters based on the relative byte address of a commit record. For *comparison_operator*, see “Comparison Operators”.

**CLEAR {filter_spec | ALL}**
Removes filtering criteria.

- **ALL** removes all filter criteria.
- **filter_spec** removes only the specified criterion. Specify any FILTER option, but not the value. The following example is valid:

  FILTER CLEAR STRING

  The following example is not valid:

  FILTER CLEAR STRING "Denver"

  An alias for CLEAR is RESET.

**CSN | LogCSN [comparison_operator] [value]**
Filters based on a commit sequence value.
For *comparison_operator*, see “Comparison Operators”.

**ENDTIME time_string**
Ends the filter at the last record written at the specified time. For the time string, use the format of:

[[yy]yy-mm-dd] [hh:mm[:ss]]

Example:

ENDTIME 2011-01-31 23:59:59

This command can be shortened to ENDS or END.
FILENAME name [, name]
Filters based on the name of a SQL table, or a group of names, with the name format being:

[catalog.]owner.table
[catalog.]owner.string

Also filters on the name of a NonStop data file, or a group of names, with the name format being:

volume.subvolume.file
volume.subvolume.string

On Windows and UNIX, both types of objects are supported. On NonStop, only viewing NonStop files is supported. To view SQL tables on NonStop, use the ANSINAME option.

FILENAME is case-sensitive on Windows and UNIX systems. If the database requires quote marks around a name. If opening a NonStop data file on Windows or UNIX, the file name is converted to upper case. FILENAME can be shortened to FILE or FI.

Up to eight name specifications may be supplied.

GGSTOKEN token_name [comparison_operator][token_value]
Filters based on a specific token in the record header.

• For token_name, provide the name of the token, such as TRANSID.
  To view a list of possible tokens and values, see “GGSTOKEN”.

• For token_value provide the actual value for this token that is to be the filter string.

For comparison_operator, see “Comparison Operators”

HEX "hex_string" [byte_range] [, "hex_string" [byte_range]]
Filters based on a hex string and, optionally, a range of columns. To specify a range of columns, use the format of:

start_column:end_column

Example:

10:35

This option allows up to eight hex string and column arguments. Hex strings must be enclosed within quotes.

Hex filter strings must be in double quotes.

INCLUDE
Specifies that the filter will include the information specified with other options in the current FILTER statement. Can be shortened to INC.

EXCLUDE
Specifies that the filter will exclude the information specified with other options in the current FILTER statement. Can be shortened to EXC.

INT16 16_bit_integer
Filters based on a 16-bit integer. Use with 16-bit processors.
**INT32 32_bit_integer**
Filters based on a 32-bit integer. Use with 32-bit processors.

**IOTYPE operation_type [, operation_type]**
Filters based on the type of operation. A list of record types can be viewed with the `SHOW RECTYPE` command in Logdump. Up to 32 operation types can be specified with IOTYPE.

**MATCH {ANY | ALL}**
Controls filtering response when multiple filters have been specified. Can be shortened to MAT or MA.

- **ANY** includes a record for display or counts if the condition matches any of the filter conditions. This is the default.
- **ALL** includes a record for display or counts only if the condition matches all of the filter conditions.

**OFF**
Disables record filtering. By default, filtering is disabled. An alias for this option is DISABLE.

**ON**
Enables record filtering. An alias for this option is ENABLE.

**RBA byte_address [comparison_operator] [ ...]**
Filters based on a relative byte address. Accepts either a 32-bit or 64-bit value. Up to 32 specifications can be supplied.

**RECLLEN length [comparison_operator]**
Filters based on a record length, in bytes. For comparison_operator, see “Comparison Operators”.

**RECTYPE {type_number | type_name}**
Filters based on the type of record. Can be either of the following:

- The number assigned to the record type.
  
  FILTER RECTYPE 10

- The name of the record type.
  
  FILTER RECTYPE Update

To view the record type names and numbers, issue the SHOW RECTYPE command. (See “SHOW”.)

**SHOW**
Displays filter settings. Same as using FILTER without any options.

**STARTTIME time_string**
Starts the filter with the first record written at the specified time. For the time string, use the format of:

```
[[yy]yy-mm-dd] [hh:mm[:ss]]
```

Example:

STARTTIME 2011-01-01 00:00:00
Can be shortened to **STARTTS** or **START**.

**STRING [BOTH] [B], ”text” [column_range] [[B], text [column_range]] […]**

- *text* filters based on a string. Enclose the string within double quotes.
- *column_range* filters based on a range of columns. Use the format of:
  
  \[start_column:end_column\]

**Example:**

10:35

- **BOTH** filters on both a string and a column range.
- **[B]** specifies a case-insensitive match. You can match up to eight string and column arguments.

If the trail data is EBCDIC, issue the **EBCDICDATA ON** or **ASCIIDATA OFF** command before using **FILTER STRING** to ensure the correct matching.

**STRING** can be shortened to **STR**.

**SYSKEY system_key [comparison_operator] […]**

Filters based on a NonStop source key. Accepts either a 32-bit or 64-bit value. Up to 32 specifications can be supplied.

**TRANSIND indicator [comparison_operator]**

Filters based on the TransInd field of the record header. Valid values:

0 = start of transaction
1 = middle of transaction
2 = end of transaction
3 = only record in transaction

For example, to filter for the end of a transaction, use the following command, including the spaces in the syntax:

FILTER INCLUDE TransInd >= 2

For *comparison_operator*, see “Comparison Operators”

**TRANSID 'transaction_identifier'**

Filters on the TMF transaction identifier when reading a TMF trail, for example:

FILTER INCLUDE TRANSID \GGQA(2).0.12792182.

**UNDOFLAG type [comparison_operator]**

Filters based on the NonStop undo flag. The undo flag is set for records that are undone when a TMF transaction is aborted. Normally, **UndoFlag** is set to zero, but if the record is the backout of a previously successful operation, then **UndoFlag** will be set to 1. An undo that is performed by the disc process because of a constraint violation is not marked as an undo.

For *comparison_operator*, see “Comparison Operators”

**USERTOKEN token_name [comparison_operator] [token_value]**

Filters based on a specific user token in the trail file header.
• *token_name* is the name of any token that is defined with the `TOKENS` clause of a `TABLE` statement of the Extract parameter file. It is not case-sensitive.

• *token_value* is either a constant that is enclosed within double quotes or the result of a column-conversion function, depending on what was specified in the `TOKENS` clause for *token_name*.

• For *comparison_operator*, see “Comparison Operators”

The following shows filter options modified by comparison operators.

```
FILTER INCLUDE RECLLEN > 400
FILTER INCLUDE RECLLEN < 200
FILTER INCLUDE TRANSEND <> 1
FILTER INCLUDE SYSKEY > 20217270057313
```

The following filters for a data file name and for a relative key 19446, which has a hex value of 00004bf6. Because `MATCH ALL` is used, a record must meet all of the filter specifications to be included in the filter.

```
FILTER INCLUDE FILENAME $QA01.QAESRC.ACCT*
FILTER INCLUDE HEX "00004bf6" 0:3
FILTER MATCH ALL
```

The following filters for tables that start with ACC except for the ACCDET table, and for records that contain a timestamp between the specified start and stop times. By default, if a record matches any of the `INCLUDE` specifications, it is included in the filter.

```
FILTER INCLUDE FILENAME SALES.ACC*
FILTER EXCLUDE FILENAME SALES.ACCDET
FILTER INCLUDE STARTTIME 2011-01-11 17:00:00
FILTER INCLUDE ENDTIME 2011-01-11 19:00:00
```

The following shows filter options with multiple specifications. By default, a record that matches any of these specifications will be included in the filter. Note that in the STRING filter, two of the criteria are not case-sensitive, while one is, and the filter is confined to a column range.

```
FILTER INCLUDE IOTYPE insert,update,delete
FILTER INCLUDE STRING b"String1" "string2" b"String3" 25:50
FILTER INCLUDE FILENAME $QA01.QAESRC.ACCT1, $QA01.QAESRC.ACCT2, $QA01.QAESRC.ACCT3
```

### 2.26 FLOAT

Use `FLOAT` to display a number or hex string in both its hex representation and as a floating-point number. This command is useful when looking for a specific floating-point number and you need the hex representation of that number. This command does not require a trail file to be opened.

The output of this command is `hex_value float_string`.

`FLOAT` assumes the floating point representation is the default for the current platform. It will not make a conversion between a Tandem floating point number (which is not IEEE-754) and an IEEE-754 floating point number, which is supported on UNIX and Windows systems.

**Default**

None
Syntax

FLOAT input_string [format]

**input_string**
Can be one of the following:

- A valid floating point number as a string in the format of the following: an optional sign character (+ or -), followed by a sequence of decimal digits, which can contain a decimal-point followed by an exponent (an e or E character, followed by an optional sign and a sequence of digits).
- A valid hexadecimal number as a string formed by the following: 0x, %H, %h, X, x, H or h, followed by a sequence of hexadecimal digits (1-9, a-f).

**format**
Can be one of the following:

- One of the following to specify the output format sizing: IEEE, TDM, TANDEM, NSK, 64bit, 64-bit, F64, 32bit, 32-bit or F32.
- A format specifier formed by the string FMT followed by any valid C99 Print Format specifier (for example, %lx, %e, %g).

**Example**
The following examples are different results that can be obtained for the same values 2.1 and 2.2.

Logdump 58 >float 2.1
4000cccccccccccd  2.100000
Logdump 59 >float 2.2
400199999999999a  2.200000
Logdump 60 >float 0x4000cccccccccccd
4000cccccccccccd  2.100000
Logdump 61 >float 0x400199999999999a
400199999999999a  2.200000
Logdump 62 >float %H400199999999999a
400199999999999a  2.200000
Logdump 63 >float 2.2 FMT %e
4001999999999999a  2.200000e+00
Logdump 64 > float 2.2 FMT %g
4001999999999999a  2.2
Logdump 65 >float 0x4000cccccccccccccd FMT %g
4000cccccccccccd  2.1
Logdump 66 >float 2.2e+01
4036000000000000  22.000000
Logdump 67 >float 2.2e-05
3ef711947cfa26a2  0.000022

2.27 GGSAUDITREAD

Use GGSAUDITREAD to read the TMF audit trail. GGSAUDITREAD bypasses TMFARLIB and causes Logdump to open and read the TMF trail directly. This command is strictly for use by support analysts.
2.28 GGSTOKEN

Use GGSTOKEN to include tokens in the record display. A token is internal information that the application places in the record for its own use. Tokens are not for use by users.

Without arguments, GGSTOKEN displays the status of token display (ON or OFF).

Tokens on NonStop are:

"TRANSID"
"BEGINSEQNO"
"BEGINRBA"
"BEGINTIMESTAMP"
"PARENTNODE"
"PROCESSNAME"
"PROGRAMNAME"
"HOMENODE"
"CHILDLIST"

Tokens on Windows and UNIX are:

"ORAROWID"
"TRANID"
"MARKER"
"FETCHEDDATA"
"FORCEDCOMMIT"
"DDL"
"LOGCSN"
"GROUPNAME"
"VAMSEQID"

Default

OFF

Syntax

GGSTOKEN {ON | OFF | DETAIL}

ON
Enables the display of tokens.

OFF
Disables the display of tokens.

DETAIL
Adds the token ID and length to the display.

Examples
Example 1

GGSTOKEN ON

This shows something similar to the following:

GGS tokens:
5200 1400 4141 4148 6561 4141 4641 4141 4141 4F41 : R...AAAHeaAAFAAAAA0A
4143 0001 : AC..

Example 2

GGSTOKEN DETAIL

This shows something similar to the following:

GGS tokens:
TokenID R (x52), Info 0, Length 20
4141 4148 8561 4141 4641 4141 4141 4F41 4144 0001 : AAAHeaAAFAAAAA0AAD..

2.29 GGSTOKEN

Valid for Oracle.

Use GGSTOKEN to control whether or not automatically generated token data is displayed with each record. It applies to the automatically generated tokens like the transaction id, row id, fetching status and tag value. These values are stored in the record header and can be mapped to a target column or used for some other purpose during processing.

Without arguments, GGSTOKEN displays the status of user token display (ON or OFF). With the ON option, the name of the token and its length are displayed. The DETAIL option shows the actual token data.

Default
Display token name and length.

Syntax

GGSTOKEN {ON | OFF | DETAIL}

ON
Enables the display of automatically generated tokens.

OFF
Disables the display of automatically generated tokens.

DETAIL
Displays the automatically generated tokens including the transaction ID (XID), the row id for DML operations, the fetching status (if applicable), and tag value.token data.
2.30 GHDR

Use GHDR to control whether or not the record header is displayed with each record. Each record contains a header that includes information about the transaction environment. Without arguments, GHDR displays the status of header display (ON or OFF).

Default
OFF

Syntax
GHDR {ON | OFF}

2.31 HASHCLEAR

Use HASHCLEAR to release memory allocated to the hash list that is generated by the HASHSTATS command.

An alias for HASHCLEAR is HASHRESET.

Default
None

Syntax
HASHCLEAR

2.32 HASHSTATS

Use HASHSTATS to enable or disable the display of statistics about file name hashing after a COUNT DETAIL command. The command without arguments displays whether the hash statistics are enabled or disabled. Use the HASHCLEAR or HASHRESET command to clear the memory allocated to the hash list.

Default
OFF

Syntax
HASHSTATS {ON | OFF}

Example
The following is the result when HASHSTATS is enabled. It appears at the end of the COUNT DETAIL display.

Files    7, Coll   0, Chain   0
Avg Hash lookup time  5 across 1093 lookups
2.33 HEADERTOKEN

Use HEADERTOKEN to control whether or not header token indicators are displayed with each record. The header token indicators are the following:

G — record header (begin of record)
H — header area
D — data area
T — internal token
U — user token area (does not display if user tokens are not in use)
Z — end of record

Without arguments, HEADERTOKEN displays the status of header token indicators (ON or OFF).

Default

OFF

Syntax

HEADERTOKEN {ON | OFF | DETAIL}

ON
Enables the display of header tokens.

OFF
Disables the display of header tokens.

DETAIL
Provides detailed token values.

Examples

Example 1
HEADERTOKEN, without DETAIL

TokenID G, Info 0, Length 117
TokenID H, Info 0, Length 45
TokenID D, Info 0, Length 28
TokenID T, Info 0, Length 24
TokenID Z, Info 0, Length 117

Example 2
HEADERTOKEN with DETAIL

TokenID G, Info 0, Length 146
TokenID H, Info 0, Length 42
4504 0041 3C00 05FF 402F AE6C 572A F102 F818 8F02 : E...A<...@/.1\......
0000 0000 1000 0000 0152 0000 0001 4852 2E4A 4F42 : ..........R....FR.JOB
5300
2.34 HELP

Use HELP to view the syntax of Logdump commands.

Default
None

Syntax
HELP

2.35 HISTORY

Use HISTORY to view the most recently issued Logdump commands since the session started, or to reset the command count starting at line 1 again. HISTORY can be shortened to HIST.

Note:
You can use the FC command to re-execute a command in the list. See “FC”

Default
Display recent commands

Syntax
HISTORY [n] [CLEAR]

n
Returns the specified number of previously issued commands, where n is any positive number.

CLEAR
Deletes the command history buffer and reverts the command line to 1.

Example
HISTORY 3

The results of this command would be similar to:

1: ghdr on
2: detail on
3: scanforheader

2.36 INTERPRETINTERVAL

Use INTERPRETINTERVAL to display a 64-bit Julian time interval in the format of days-hh:mm:ss.ms.us.
**INTERPRETTIMESTAMP**

Use **INTERPRETTIMESTAMP** to display a 64-bit Julian timestamp as an ASCII value.

**Default**
None

**Syntax**

```
INTERPRETTIMESTAMP timestamp
```

*timestamp*
A JULIANTIMESTAMP value.

**Example**

```
INTERPRETTIMESTAMP 211976584185800569
```

This produces the following result:

```
2005/03/03 04:29:45.800.569 GMT
2005/03/02 20:29:45.800.569 LCT
```

**2.38 LOG**

Use **LOG** to start and stop the logging of Logdump sessions. When enabled, logging remains in effect for all sessions of Logdump until disabled with the **LOG STOP** command. Without arguments, **LOG** displays the status of logging (ON or OFF). An alias for **LOG** is **OUT**.

**Default**

Disabled
Syntax

\texttt{LOG \{file\_name | STOP\}}

\texttt{file\_name}
Specifies the name of the log file. Specify a full path name to store the file in a directory other than the current working directory.

\texttt{STOP}
Stops logging.

Example

\texttt{LOG /home/ggs/dirrpt/logdumpout.txt}

\texttt{Example}

\texttt{LOG $data01.glogggl.sesslog}

2.39 Metadata \texttt{ddlformat}

Use Metadata \texttt{ddlformat} on a NonStop system to display the TDR columns in Nonstop DDL format.

Default

ON

Syntax

\texttt{Metadata \texttt{ddlformat} \{ON | OFF\}}

2.40 Metadata \texttt{defgenformat}

Use Metadata \texttt{defgenformat} on a NonStop system to display the columns in a TDR in the format of a GG definition from a definition file.

Default

OFF

Syntax

\texttt{Metadata \texttt{defgenformat} \{ON | OFF\}}

2.41 Metadata \texttt{detail}

Use Metadata \texttt{detail} on a NonStop system to display the tokens and values that make up a metadata.

Default

OFF
2.42 Metadata sqlformat

Use Metadata sqlformat on a NonStop system to display the columns in a TDR in the format of a SQL table create.

Default
OFF

Syntax
Metadata sqlformat  {ON | OFF | DATA}

2.43 NEXT

Use NEXT to display the next record or records in the file. The default displays only the next record. NEXT can be shortened to N. An alias for NEXT is RECORD.

Default
Display the next 1 record

Syntax
NEXT [n]

n
Displays the specified number of subsequent records.

Example
NEXT 10

2.44 NEXTTRAIL

Use NEXTTRAIL to close an open trail file and open the next one in the sequence. An alias for NEXTTRAIL is NT.

Default
None

Syntax
NEXTTRAIL

2.45 NOTIFY

Use NOTIFY to display the number of records scanned, the trail position, and the record timestamp at specified intervals when using COUNT and records are being suppressed from display through filtering options. An alias for NOTIFY is NOTIFYINTERVAL.
Instead of displaying each notify interval on a separate line, you can configure Logdump to simply update a single line with each new scan result. See “SCANSCROLLING”.

**Default**
None

**Syntax**
NOTIFY interval

**interval**
The notification interval expressed as a number of records.

**Example**
The following shows the usage and result of this command.

```
Logdump 26> NOTIFY 1000
Logdump 27> FILTER INCLUDE FILE sales.res*
Logdump 28> COUNT
Scanned 1000 records, RBA 160380, 2011/02/01 08:53:47.768.255
Scanned 2000 records, RBA 729961, 2011/02/01 08:56:09.916.128
Scanned 3000 records, RBA 2032683, 2011/02/01 08:56:09.916.128
Scanned 4000 records, RBA 3244585, 2011/02/01 08:56:09.916.128
Scanned 5000 records, RBA 4568766, 2011/02/01 08:56:09.916.128
```

### 2.46 OBEY

Use OBEY to process a file that contains a list of Logdump commands. OBEY is useful for executing commands that are frequently used in sequence.

OBEY can be shortened to O. An alias for OBEY is SOURCE.

**Default**
None

**Syntax**
OBEY file_name

**file_name**
The fully qualified name of the file containing the list of commands.

**Example**
This is a UNIX example.

```
OBEY ./ldcommands.txt
```

**Example**
This is a NonStop example.

```
OBEY $DATA01.GGSPARM.OBEY1
```
The preceding command executes a file that might look something like this:

```plaintext
ghdr on
usertoken on
detail
filter enable
filter clear
filter match all
```

### 2.47 OPEN

Use **OPEN** to open a trail file or extract file in Logdump. Without arguments, the command displays the name of the file that is currently open. Aliases for **OPEN** are **FROM** and **LOGTRAIL**.

**Default**

None

**Syntax**

```plaintext
OPEN file_name
```

*file_name*

The fully qualified path name of the trail file or extract file to be opened. To specify a trail file, specify the trail name (a two-character prefix) and the sequence number, for example `jd000000`.

**Example**

This is a UNIX example.

```
OPEN /home/ggs/dirdat/jd000000
```

**Example**

This is a NonStop example.

```
OPEN $data01.glogggl.aa000000
```

### 2.48 POSITION

Use **POSITION** to set the read position in the file. The position of a record in the file is noted in the record header in the `AuditPos` field.

Without options, **POSITION** displays the current read position. Options let you specify an exact position. After you set the position, issue the **NEXT** command to view the record at that position.

**POSITION** can be shortened to **POS**.

**Default**

None

**Syntax**

```plaintext
POSITION [bytes | {0 | FIRST}]
```
Specifying the number of bytes into the file at which to read. Use the NEXT command to view the specified record.

0 | FIRST
Positions Logdump at the beginning of the file.

Syntax
POS 77580548

2.49 RECLLEN

Use RECLLEN to control how much of the record data is displayed. You can use RECLLEN to control the amount of scrolling that must be done when records are large, while still showing enough data to evaluate the record. Data beyond the specified length is truncated.

Default
140 bytes

Syntax
RECLLEN n

n
The number of bytes of the record that is displayed.

Example
RECLLEN 280

2.50 SAVE

Use SAVE to write a subset of the records to a new trail or extract file. By saving a subset to a new file, you can work with a smaller file that is easier to debug. Saving to another file also enables you to extract valid records that can be processed by , while excluding records that may be causing errors.

To set the version of the trail or file (to old or new format), use the TRAILFORMAT command.

Default
None

Syntax
SAVE file_name [!] (n records | n bytes)
[NOCOMMENT]
[OLDFORMAT | NEWFORMAT]
[TRANSIND indicator]
[TRUNCATE]
**file_name**
The name of the new file. To specify a trail file, specify the two-character trail name and a sequence number, for example `rt000001`.

! Overwrites the specified file, if the same file already exists. First a purge is done, and then the specified records are saved to the file.

**n records | n bytes**
Specifies either a number of records or a number of data bytes to write to the new file. The n number of records or bytes are taken forward from the current position in the file. You can change the position with the `POSITION` command. See “POSITION”.

**EXT (pri, sec [, max])**
Specifies savefile extent sizes. This option is valid on NonStop only.

**MEGABYTES n**
Specifies the size of a savefile extent. This option is valid on NonStop only.

**NOCOMMENT**
Suppresses the leading and trailing comment records that are placed by default in the new file. These records describe the context of the file. The begin comment record contains source trail information and the position where the save started. The end comment record identifies the end of the saved data. These headers are useful to separate different sets of records that are saved to the same file, but can be omitted.

**OLDFORMAT | NEWFORMAT**
Writes the data in either the current trail format (NEWFORMAT, the default) or the format that was used for versions 6.0 and earlier (OLDFORMAT).

**TRANSIND indicator**
Sets the TransInd header field in the records written to one of the following:

- **FIRST**
- **MIDDLE**
- **END**
- **ONLY**

This allows you to reorder records in a transaction. TRANSIND applies to all records written by a `SAVE` command.

**TRUNCATE**
Purges an existing file before saving new information to it.

**Example**

SAVE /home/ggs/dirdat/rt000001 10 records nocomment

**Example**

SAVE $data01.glogggl.ss000000 100 records
2.51 SCANFORENDTRANS

Use SCANFORENDTRANS to scan for a record that has a transaction indicator of 2 or 3, as shown in the TransInd field of the header. When one of those indicators is found, Logdump displays the first record of the next transaction.

The indicators represent the following:

- 2 — last record in the transaction
- 3 — only record in the transaction

SCANFORENDTRANS can be shortened to SFET.

**Default**

None

**Syntax**

SCANFORENDTRANS

2.52 SCANFORHEADER

Use SCANFORHEADER to go to the next record header. Before using this command, use the GHDR ON command to show record headers (see “GHDR”). SCANFORHEADER can be shortened to SFH.

**Default**

None

**Syntax**

SCANFORHEADER [PREV]

PREV
Displays the previous record header.

2.53 SCANFORMETADATA

Use SCANFORMETADATA to scan for a specific metadata record. SCANFORMETADATA can be shortened to SFMD.

**Default**

None

**Syntax**

SCANFORMETADATA [DDR|TDR] [Index|NEXT]
SFMD [DDR|TDR] [Index|NEXT]
**2.54 SCANFORRBA**

Use `SCANFORRBA` to scan for the record at a relative byte address specified by the `AuditRBA` field of the record header. Before using this command, use the `GHDR` command to show record headers (see “GHDR”). `SCANFORRBA` can be shortened to `SFR`.

**Default**
None

**Syntax**

```
SCANFORRBA relative_byte_address [file_name]
```

*relative_byte_address*

Specifies the relative byte address to find.

*file_name*

Constrains the search to an Enscribe or SQL data file. A file name is required even if you are searching a file that is open in Logdump.

**Example**

```
SCANFORRBA 321 /home/ggs/dirdat/rt000000
```

**Example**

```
SCANFORRBA 321 $data01.glogggl.rt000000
```

**2.55 SCANFORTIME**

Use `SCANFORTIME` to scan for a record that contains a specific timestamp. The timestamp is contained in the IO Time field of the record header. Before using this command, use the `GHDR` command to show record headers (see “GHDR”). `SCANFORTIME` can be shortened to `SFTS`.

**Default**
None

**Syntax**

```
SCANFORTIME time_string [, name]
```
time_string
Scans for a specific timestamp. For the time string, use the format of:

\[yy\]yy-mm-dd [hh[:mm][:ss]]

name
Constrains the search to a specific table or data file name, or a group of names specified with a wildcard.

Example
SCANFORTIME 2011-01-27 14:33:57

2.56 SCANFORTYPE

Use SCANFORTYPE to scan for the next record of the specified type. SCANFORTYPE can be shortened to SFT.

Default
None

Syntax

SCANFORTYPE {type_name | type_number}

type_name | type_number
Specifies the type of record to search for, either by type name or type number. To view a list of record types and their associated numbers, use the SHOW RECTYPE command (see “SHOW”).

Example
Both of the following commands return the same result: They display commit records.

SCANFORTYPE Commit
SFT 2

2.57 SCANSCROLLING

Use SCANSCROLLING to configure Logdump to update a single line after COUNT scans when NOTIFY is enabled. Otherwise, each scan notification appears on a different line. See “NOTIFY” for more information.

Default
OFF

Syntax

SCANSCROLLING {ON | OFF}

ON
Enables the use of a single line for count notification results.
OFF
Disables the use of a single line, causing a separate line to be used for each notification.

2.58 SHOW

Use SHOW to display internal Logdump information, including files that are open if the system is NonStop, the current Logdump environment, a list of record types, and current filter settings. SHOW can be shortened to SH or SHO.

Default
None

Syntax

SHOW
[ENV]
[FILTER]
[OPEN]
[RECTYPE]

ENV
Displays the current Logdump environment. Same as the ENV command (see “ENV”).

FILTER
Displays current filter settings.

OPEN
Shows all NonStop files that are open in Logdump.

RECTYPE
Displays a list of record types that can be displayed with Logdump.

Examples

Example 1
SHOW FILTER

This shows something similar to the following on Windows or UNIX:

Data filters are ENABLED
Include Match ALL
Filename-0 : $QA01.QAESRC.ACCT*
HEX-0 : ( 4), Col 0:3
0000 4BF6
Exclude Match ANY

It shows something similar to the following on NonStop:

Data filters are ENABLED
Include Match ALL
Rectypes : Delete
SHOW

Filename-0 : hr.regions
Exclude Match ANY

Example 2
SHOW OPEN

This shows something similar to the following:

0 : $RECEIVE
1 : \GGS2.$ZTN2.#PTW6EUX
2 : \GGS2.$DATA4.#0009047
3 : \GGS2.$ZTN2.#PTW6EUX
4 : \GGS2.$DATA4.CPSDAT.TM000000

Example 3
SHOW RECTYPE

This shows results similar to the following. (This list might not reflect all possible record types. New types are added when needed to support new functionality.)

1 - Abort
2 - Commit
3 - Delete
4 - EndRollBack
5 - Insert
6 - Prepared
7 - TMF-Shutdown
8 - TransBegin
9 - TransRelease
10 - Update
11 - UpdateComp
12 - FileAlter
13 - FileCreate
14 - FilePurge
15 - FieldComp
16 - FileRename
17 - AuxPointer
18 - NetworkCommit
19 - NetworkAbort
20 - CurrentPos
89 - SQL/MX DDL OP
90 - GGSSQLCol
100 - GGSPurgedata
101 - GGSPurgeFile
102 - GGSCreateFile
103 - GGSAlterFile
104 - GGSRenameFile
105 - GGSSetmode
107 - GGSControl
106 - GGSChangeLabel
2.59 SKIP

Use SKIP to skip the specified number of records.

Default
None

Syntax
SKIP \( n \)

\( n \)
The number of records to skip.

Example
SKIP 50

2.60 TIME

Use TIME to display the current time in local and GMT formats.

Default
None
2.61 TIMEOFFSET

Use `TIMEOFFSET` to set the Logdump time format. Without arguments, `TIMEOFFSET` displays the current time offset. Options enable you to set the time to the local time, Greenwich Mean Time (GMT), or a specific offset from GMT. The specified time format applies to the timestamps shown in records as well as any Logdump commands that accept a time string argument.

Default

LOCAL

Syntax

```
TIMEOFFSET {LOCAL | GMT | GMT + hh[:mm] | GMT - hh[:mm]}
```

**LOCAL**

Sets the time to that of the local system.

**GMT**

Sets the time to Greenwich Mean Time (GMT).

**GMT + hh[:mm]**

Sets the time ahead of GMT by the specified number of hours and, optionally, minutes.

**GMT - hh[:mm]**

Sets the time behind GMT by the specified number of hours and, optionally, minutes.

Example

```
TIMEOFFSET GMT -01
```

2.62 TMFBEFOREIMAGE

Use `TMFBEFOREIMAGE` on a NonStop system to view the before image for update operations from TMF audit.

Default

OFF

Syntax

```
TMFBEFOREIMAGE {ON | OFF}
```

**ON**

Displays the before image for update operations from the TMF audit.

**OFF**

Displays only the after image for update operations from the TMF audit.
Example
A sample display for TMFBEFOREIMAGE ON is shown below.

```
2011/01/12 10:02:34.325.264 FieldComp          Len    38 RBA 615854956
Name: \NY.$DATA1.GGSDAT.TCUSTMER
Before Image:                                             Partition 0
0000 0004 414E 4E20 0002 0014 5345 4154 544C 4520 | ....ANN ....SEATTLE
2020 2020 2020 2020 2020 0003 0002 5741      | ....WA
2011/01/12 10:02:34.325.264 FieldComp          Len    38 RBA 615854956
Name: \NY.$DATA1.GGSdat.TCUSTMER
After  Image:                                             Partition 0
TRANSID     : \NY(2).0.7022034  (7998393398406021122)
0000 0004 414E 4E20 0002 0014 4E45 5720 594F 524B | ....ANN ....NEW YORK
2020 2020 2020 2020 2020 0003 0002 4E59      |             ....NY
```

2.63 TMFBEFOREIMAGE

Use TMFBEFOREIMAGE on a NonStop system to set the TMFARLIB to fetch the before image of the record and display it with the after image. Without an argument, this command displays whether the fetching of before images is on or off.

Default
OFF

Syntax
TMFBEFOREIMAGE {ON | OFF}

2.64 TMFGETRECADDR | NOTMFGETRECADDR

Use TMFGETRECADDR and NOTMFGETRECADDR on a NonStop system to control the ability of Logdump to call the ARGETRECADDR() function of TMFARLIB. This function is used when examining a TMF audit trail.

Default
None

Syntax
TMFGETRECADDR | NOTMFGETRECADDR

**TMFGETRECADDR**
Enables the use of the ARGETRECADDR() function.

**NOTMFGETRECADDR**
Disables the use of the ARGETRECADDR() function.

2.65 TMFIGNORERECCOUNT

Use TMFIGNORERECCOUNT on a NonStop system to set the number of records that the TMFARLIB can ignore before returning a CURRENTPOSITION record.
2.66 TRAILFORMAT

Use TRAILFORMAT to set the version of the trail or extract file that is being saved when using the SAVE command.

Default
None

Syntax

```plaintext
TMIGNOREREACCOUNT n
```

2.67 TRANSBYTELIMIT

Use TRANSBYTELIMIT to prevent normal-sized transactions from being tracked in the transaction table specified with the TRANSHIST command. It sets a lower boundary for the number of bytes in a transaction and should be set to represent a normal-sized transaction for the environment being evaluated with Logdump. Setting a boundary reduces the amount of data that is stored and, consequently, the amount that must be reviewed when troubleshooting.

Default
10000 bytes

Syntax

```plaintext
TRANSBYTELIMIT n
```

n
The number of bytes in a normal-sized transaction.

Example

```plaintext
TRANSBYTELIMIT 9000
```

2.68 TRANSHIST

Use TRANSHIST to keep track of the size of transactions in a trail or file. Logdump tracks the transactions in an internal history table in descending order according to the
number of bytes of data in each one. When the history table is full, the smallest transaction is removed to allow a larger transaction to be added to the list.

Use `TRANSHIST` in conjunction with other Logdump commands to determine whether or not your applications generate large transactions and to identify their relative size. This information can be used when deciding how to group tables into different processing groups for faster throughput. For more information, see “Evaluating Transaction Size”.

Note:
You can use the `SEND EXTRACT` command with the `SHOWTRANS` option to view a list of long-running transactions. Other options enable you to control whether those transactions are ignored or processed by Oracle GoldenGate.

**Default**
0 (do not maintain history)

**Syntax**

```
TRANSHIST n
```

**Example**

```
TRANSHIST 150
```

## 2.69 TRANSRECLIMIT

Use `TRANSRECLIMIT` to prevent normal-sized transactions from being tracked in the transaction table specified with the `TRANSHIST` command. It sets a lower boundary for the number of records in a transaction and should be set to represent a normal-sized transaction for the environment being evaluated with Logdump. Setting a boundary reduces the amount of data that is stored and, consequently, the amount that must be reviewed when troubleshooting.

**Default**
100 operations

**Syntax**

```
TRANSRECLIMIT n
```

**Example**

```
TRANSRECLIMIT 90
```
2.70 **USERTOKEN**

Use **USERTOKEN** to control whether or not user token data is displayed with each record. A user token is data specified by an user that is stored in the record header and can be mapped to a target column or used for some other purpose during processing.

Without arguments, **USERTOKEN** displays the status of user token display (ON or OFF). With the **ON** option, the name of the token and its length are displayed. The **DETAIL** option shows the actual token data.

**Default**
Display token name and length.

**Syntax**

```
USERTOKEN {ON | OFF | DETAIL}
```

**ON**
Enables the display of user tokens.

**OFF**
Disables the display of user tokens.

**DETAIL**
Displays the token data.

2.71 **VOLUME**

Use **VOLUME** to set the default directory, volume or subvolume. An alias for this command is **CD**.

**Default**
None

**Syntax**

```
VOLUME {directory | volume | subvolume}
```

2.72 **WRITELOG**

Use **WRITELOG** to write text to the session log. Before using this command, start logging with the **LOG** command (see “**LOG**”).

**Default**
None

**Syntax**

```
WRITELOG text
```
**text**
Any text string. Quotes are optional.

**Example**
WRITELOG "Customer name is ABC Company."

### 2.73 x

Use `x` to execute a program from within Logdump. When you exit the program, the Logdump prompt returns.

**Default**
None

**Syntax**

```
x program [string]
```

**command**
The program to run.

**string**
A character string, such as input arguments.

**Example**
The following series of commands and output shows how you can exit Logdump, issue other commands from the shell or within GGSCI, and then return to the Logdump command line.

Logdump 696 >x ggsci

GoldenGate Command Interpreter
Version ..... 

```
GGSCI (sysa) 1> status er *
GGSCI (sysa) 2> start er *
GGSCI (sysa) 3> info er *
GGSCI (sysa) 4> exit
Logdump 697 >
```
Index

A

audit trail
  TMF
    reading, 2-33

G

GGAUDITREAD command
  Logdump, 2-33
GGSTOKEN command
  Logdump, 2-34

H

HASH commands
  Logdump, 2-36
HASHSTATS command
  Logdump, 2-36

M

memory
  (continued)
    releasing from Logdump hash, 2-36

S

statistics
  file name hashing, 2-36

T

TMF audit trail
  reading, 2-33
tokens
  internal
    Oracle GoldenGate, 2-34

V

viewing
  tokens, 2-34