

Using ClustrixDDB as a Replication Master

- [About Binary Logs](#)
- [MySQL 5.7 Replication and GTIDs](#)
- [Create Replication User](#)
- [Creating a Binary Log File](#)
- [Setting Binary Logging Options](#)
- [Displaying Binary Log Information](#)
- [Trimming a Binary Log](#)
- [Backing Up Binary Logs](#)
- [Excluding A Session from Binary Logs](#)
- [Dropping a Binary Log File](#)
- [Global Variables](#)
- [Related Links](#)

About Binary Logs

ClustrixDDB implements MySQL compatible binlogs that with the following additional features:

- ClustrixDDB can maintain multiple independent binary logs (for example, binlogs per table, per database, or for a set of tables and databases).
- The binary logs are fault-tolerant, with the same guarantees as the rest of the ClustrixDDB system.
- Writes to the binlog are transactional, consistent, and durable (full ACID guarantee).
- Binary logs can be created and dropped online.

To configure a ClustrixDDB system with a single row-based binary log, issue the following commands:

```
master> CREATE BINLOG 'clustrixDDB-bin' FORMAT='ROW';
master> SHOW MASTER STATUS;
```

To disable binary logging and drop (permanently) an existing binlog:

```
master> DROP BINLOG 'clustrixDDB-bin';
```

When running MySQL database as a slave to a ClustrixDDB master, ClustrixDDB does not support the variable `binlog_checksum`, which causes the master to write checksums for events written to the binary log.

MySQL 5.7 Replication and GTIDs

MySQL produces global transaction identifiers (GTIDs) beginning with MySQL 5.6 (optional) and MySQL 5.7 (required). ClustrixDDB does not implement nor support GTIDs. To enable replication between ClustrixDDB and MySQL with GTID, use the following settings:

For ClustrixDDB (Master) to MySQL 5.7 (Slave) Replication

- The MySQL slave must have the global `gtid_mode` set to `OFF`, `OFF_PERMISSIVE`, or `ON_PERMISSIVE`.
- The MySQL startup option `enforce-gtid-consistency` should be set to `OFF` on the slave. See [Startup Options Used with GTID Replication](#).

For MySQL 5.7 (Master) to ClustrixDDB (Slave) Replication

- No special configuration is needed, but ClustrixDDB will ignore values for `gtid` and `server_uuid`
- Further information is available at [Using ClustrixDDB as a Replication Slave](#).

ClustrixDDB does not pass GTID events to its binlogs. This is similar to the behavior of a MySQL 5.6 slave with `gtid_mode` set to `OFF`.

Create Replication User

The user name and password used for replication are stored as plain text within the binlogs. As such, ClustrixDDB recommends establishing a separate account for exclusive use with replication to prevent compromising the security of regularly used accounts.

Follow this sample to create an account that will be used when setting up a slave. You must have privileges to `CREATE USER` and `GRANT` to perform this step.

```
master> CREATE USER 'replication'@'%' IDENTIFIED BY 'clustrixDDB';
master> GRANT REPLICATION SLAVE ON *.* TO 'replication'@'%';
```

Creating a Binary Log File

To create a binary log, issue the following command:

```
CREATE BINLOG Syntax
```

```
master> CREATE BINLOG '
binlog_name' [LOG (target1, target2, ...),]
[IGNORE (target3, target4, ...),]
[FORMAT= 'STATEMENT' | 'ROW']
```

Example

```
master> CREATE BINLOG 'mybinlog' FORMAT='ROW';
```

Optional attributes are:

- LOG: A list of specific tables or databases to log
- IGNORE: A list of specific tables or databases to ignore
- FORMAT: Format specifier (STATEMENT or ROW).

By default, this command creates a binary log file for the entire cluster in STATEMENT format, which is the most common format in MySQL environments. Alternatively, you can create binlogs that scope a database or a list of tables. For more information, please see the section on [Binlog scope](#).

For most workloads, row-based replication (FORMAT='ROW') provides better performance than statement-based replication. If you are unsure what is most appropriate for your environment, contact [Clustrix Support](#).

If a database is set to both LOG and IGNORE, ClustrixDB will IGNORE. This deviates from MySQL, which will log and not ignore.

Setting Binary Logging Options

To configure binary logging options, issue the ALTER BINLOG command. Options for the ALTER BINLOG logfile command are as follows.

Flag	Description
LOG (db1, db2)	Only log updates to databases db1 and db2
IGNORE (db3)	Ignore updates to db3
ADD LOG (db4)	Log updates to db4, in addition to others
ADD IGNORE (db5)	Ignore updates to db5, in addition to others
DROP LOG (db6)	Stop logging to db6
LOG ALL	Log updates to all databases, as opposed to specific databases. Does not reset the IGNORE list.
DISABLE	Disable logging to this binlog
ENABLE	Enable logging to this binlog
RENAME bar	Rename specified binlog to "bar"
FORMAT='row' or 'statement'	Configure log format (row-based or statement-based)

Displaying Binary Log Information

If only one binary log exists, you can display its filename, segment number and position by issuing the following command:

```
master> SHOW MASTER STATUS;
```

If more than one binary log exists, the log configured by the global variable `master_status_binlog` is displayed. If `master_status_binlog` is unset, an error is returned. This behavior is compatible with behavior of the MySQL `mysqldump --master-data` command.

```
master> SET GLOBAL master_status_binlog = 'foo';
master> SHOW MASTER STATUS;
```

To display status for all binary logs, issue the following command:

```
master> SHOW ALL MASTER STATUS;
```

To display detailed information about binary logs, issue the following command:

```
master> SHOW BINLOGS;
```

Most of this information is not directly useful, though log size can help you decide whether to trim the log.

Trimming a Binary Log

You can trim a binary log using either of the following methods:

- TRIM BINLOG command
- trim-binlog script

Trimming using the TRIM BINLOG Command

Back up your database regularly using the `mysqldump --master-data` command, which records the binary log filename at the start of the dump. To keep the size of the binary log under control, use this value to trim older data after it is backed up. The extent to which you trim is a matter of policy: you can choose to retain a week's history, or you might prefer to minimize disk consumption as much as possible by trimming all but the current file. To minimize the amount of space being used by your binary log, trim according to the Slave that is farthest behind in replication.

To list the files that compose the binary log, issue the following:

```
master> SHOW BINLOG FILES;
+-----+-----+-----+
| File           | Size      | First Event Timestamp |
+-----+-----+-----+
| eukanuba.000001 | 104857600 | 2016-01-09 19:51:08   |
| eukanuba.000002 | 104857600 | 2016-01-09 20:02:09   |
| eukanuba.000003 | 104857600 | 2016-01-09 22:22:27   |
| eukanuba.000004 | 104857600 | 2016-01-09 22:30:37   |
| eukanuba.000005 | 104857600 | 2016-01-09 22:38:11   |
| eukanuba.000006 | 104857600 | 2016-01-09 22:45:44   |
| eukanuba.000007 | 104857600 | 2016-01-09 22:53:03   |
| eukanuba.000008 | 104857600 | 2016-01-09 23:00:44   |
| eukanuba.000009 | 104857600 | 2016-01-09 23:07:46   |
| eukanuba.000010 | 104857600 | 2016-01-09 23:15:00   |
| ...           |           |                       |
+-----+-----+-----+
```

To display current Slave locations, issue the `SHOW SLAVE STATUS` command, which displays status as follows:

```
master> SHOW SLAVE STATUS\G
***** 1. row *****
      Slave_Name: default
      Slave_Status: Running
      Master_Host: alpo
      Master_Port: 3306
      Master_User: root
      Master_Log_File: alpo
      Slave_Enabled: Enabled
      Log_File_Seq: 3383
      Log_File_Pos: 58790712
      Last_Error: no error
      Connection_Status: Connected
      Relay_Log_Bytes_Read: 0
      Relay_Log_Current_Size: 0
      Seconds_Behind_Master: 0
1 row in set (0.00 sec)
```

The filename argument is the filename returned by the `SHOW MASTER STATUS` command. The sequence number (`Log_File_Seq`) indicates the binary log file currently in use (it's the numeric portion of the file name). To delete old binary data without dropping the entire log, issue the following command (`nnn` represents the sequence number):

```
master> TRIM BINLOG '
binlog_name' BEFORE
FILE 'binlog_name.
nnnn';
```

For example, if the `clx001.000283` file is timestamped at 7:15 PM on September 28, 2016, the following sample would delete all log files before that time.

```
master> TRIM BINLOG 'clx001' BEFORE FILE 'clx001.000283';
```

Trimming Using the binlog-trim Script

You can find the binlog-trim script in /opt/clustrix/bin/binlog-trim of your system.

Usage

binlog-trim [options]

Options

Option	Description
-h, --help	Show this help message and exit
-H HOST, --host=HOST	Specifies the host
-P PORT, --port=PORT	SQL port, default is mysql port: 3306
-u USER, --user=USER	Username, default is root
-p PASSWD, --passwd=PASSWD	Specifies the password
-n NUM_FILES, --num-files=NUM_FILES	Number of files to trim a time
-d, --dry	Do not perform any actual trims
-k KEEP_DAYS, --days=KEEP_DAYS	Keep this many days of binlogs
-i INTERVAL, --interval=INTERVAL	Seconds between trims
-b BINLOG_NAME, --binlog_name=BINLOG_NAME	Name of binlog to trim; must specify if multiple binlogs exist
-M MAX_RUN_TIME_MINS, --max-run-time-mins=MAX_RUN_TIME_MINS	Maximum time (minutes) script may run
-V, --version	Indicates the version

binlog-trim is generally deployed as a cron job on one of the nodes. An example entry to run once a day at 5:35UTC, with a retention policy of 7 days, trimming no more than 50 files at a time, with a minimum 60-second pause between trims, and operating on the binlog called clustrix-bin:

```
35 5 * * * root /bin/binlog-trim -H localhost -i 60 -k 7 -n 50 -b clustrix-bin 2>&1 >> /var/log/binlog-trim.log
```

The INTERVAL is a minimum wait between trims; there is additional logic in the script to prevent the trims from building up too much cleanup work (the logs will indicate this with 'waiting for bigc to pass trim').

Backing Up Binary Logs

Because ClustrixDB binary logs (binlogs) aren't stored as plain files, they cannot be backed up as MySQL binlogs can. For backup purposes, ClustrixDB provides the repclient utility, which copies binlogs from a ClustrixDB or MySQL system as if it were a replication slave. The repclient utility can be run on any ClustrixDB node.

To copy all of the binlogs off a ClustrixDB cluster, perform the following steps:

1. To list the most recent binlog, issue the SHOW MASTER STATUS command. The command returns a filename such as clustrix-bin.001903.
2. Create a directory in the /clustrix mount on a node and cd to it.
3. To retrieve all binlog files up to the most recent, issue the following command:

```
shell> node# repclient -addr 10.52.2.20 -dumpbinlog -logname clustrix-bin.000001 -end_logname clustrix-bin.001903
```

By default, the tool outputs decoded binlog messages to stdout. To specify an output file, specify the -dumpbinlog option. If you intend to archive the binlogs, omit -logpos, which can create gaps in the resulting binlog. By default, the utility stays connected to the master. To specify when it is to disconnect, include the -end_logname or -end_logpos option.

Valid options for the repclient command are as follows:

Flag	Description
-addr hostname	Database host (default: 127.0.0.1)
-count n	Number of messages to dump
-dumpbinlog	Dump binlog
-end_logname path	Ending replication log name
-end_logpos offset	Ending replication log position (default: EOF)
-help	List command options
-help-debug	List command options plus debugging output options

-logname path	Starting replication log name
-logpos offset	Starting replication log position (default: 4)
-max-packet-size bytes	Maximum packet size (default: 16777216)
-max-retries n	Maximum retries after an error (default: 3)
-no-decode-rows	Don't decode row values
-pass password	Database password (default: #undef)
-perf	Dump performance statistics
-perf-interval seconds	Dump performance statistics interval (default: 30)
-port port	Database port (default: 3306)
-retry-timeout seconds	Timeout in seconds for retries (default: 10)
-set-variable NAME= VALUE	Set a variable to the given value
-slave-id n	Slave ID (default: 1)
-testconnect	Test database connection and display status
-truncate	Truncate any existing files
-user username	Database username (default: root)
-verbose	Display debugging messages

Excluding A Session from Binary Logs

To prevent a session's statements from being inserted into any binary log, set `sql_log_bin` to false by issuing the following command:

```
master> SET sql_log_bin=false;
```

This variable inherits the value of the identically-named global variable at the start of each session. To replicate from a ClustrixDB instance, set `sql_log_bin` to true.

Be careful using `sql_log_bin` in production. Improper use can lead to data skew between the master and the slave(s).

Dropping a Binary Log File

To stop logging to the specified binary log and drop it from the system, issue the following command:

```
master> DROP BINLOG binlog_name;
```

You cannot recover a binary log after dropping it.

Global Variables

The following global and session variables control binary log behavior:

Name	Description	Default Value	Session Variable
<code>binlog_checksum</code>	Always NONE. Clustrix masters do not support generating event checksums.	NONE	
<code>binlog_format</code>	Force all binlogs to log in this format, unless set to 'DEFAULT'.	DEFAULT	✓
<code>gtid_mode</code>	Always OFF. Clustrix masters do not support generating GTID events.	OFF	
<code>gtid_purged</code>	Dummy variable for compatibility. (Clustrix does not support replication with Global Transaction Identifiers.)		
<code>master_status_binlog</code>	Binlog used in SHOW MASTER STATUS when used without specifying a binlog.		✓
<code>sql_log_bin</code>	Log statements to binary logs. This variable can be set to FALSE on a per-session basis.	true	✓
<code>sync_binlog</code>	Dummy variable for compatibility.	0	

Exercise extreme care when changing these settings. The defaults may not be ideal for your system, but they should be reasonable. The product will not warn you if you configure inadvisable settings.

Related Links

The following pages describe areas that should be understood when using ClustrixDB as a Replication Master

1. [Online Schema Changes](#)
2. [Replicating User Account Management Statements](#)
3. [Using ClustrixDB with Multiple Slaves](#)