

---

# MySQL Enterprise Backup 3.12 Release Notes

## Abstract

This document lists the changes to the MySQL Enterprise Backup 3.12 product, beginning with the most recent release. Each release section covers added or changed functionality, bug fixes, and known issues, if applicable. For information about changes in a different MySQL Enterprise Backup series, see the release notes for that series.

For additional MySQL Enterprise Backup 3.12 documentation, see the [MySQL Enterprise Backup User's Guide \(Version 3.12.5\)](#).

For legal information, see the [Legal Notices](#).

For help with using MySQL, please visit the [MySQL Forums](#), where you can discuss your issues with other MySQL users.

Document generated on: 2023-02-28 (revision: 26199)

## Table of Contents

Preface and Legal Notices .....	1
Changes in MySQL Enterprise Backup 3.12.5 (2020-01-24) .....	3
Changes in MySQL Enterprise Backup 3.12.4 (2019-02-15) .....	3
Changes in MySQL Enterprise Backup 3.12.3 (2016-05-09) .....	5
Changes in MySQL Enterprise Backup 3.12.2 (2016-01-21) .....	7
Changes in MySQL Enterprise Backup 3.12.1 (2015-06-29) .....	8
Changes in MySQL Enterprise Backup 3.12.0 (2015-03-16) .....	9
Index .....	11

## Preface and Legal Notices

This document lists the changes to the MySQL Enterprise Backup 3.12 product, beginning with the most recent release.

### Legal Notices

Copyright © 2003, 2023, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

This documentation is NOT distributed under a GPL license. Use of this documentation is subject to the following terms:

You may create a printed copy of this documentation solely for your own personal use. Conversion to other formats is allowed as long as the actual content is not altered or edited in any way. You shall not publish or distribute this documentation in any form or on any media, except if you distribute the documentation in a manner similar to how Oracle disseminates it (that is, electronically for download on a Web site with the software) or on a CD-ROM or similar medium, provided however that the documentation is disseminated together with the software on the same medium. Any other use, such as any dissemination of printed copies or use of this documentation, in whole or in part, in another publication, requires the prior written consent from an authorized representative of Oracle. Oracle and/or its affiliates reserve any and all rights to this documentation not expressly granted above.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <https://www.oracle.com/corporate/accessibility/>.

## Access to Oracle Support for Accessibility

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <https://www.oracle.com/corporate/accessibility/learning-support.html#support-tab>.

## Changes in MySQL Enterprise Backup 3.12.5 (2020-01-24)

### Bugs Fixed

- A restore operation for a TTS backup failed if the backed-up server has `ANSI_QUOTES` as one of its SQL modes, as specified in its system variable `sql-mode`. (Bug #28979134)

## Changes in MySQL Enterprise Backup 3.12.4 (2019-02-15)

- [Functionality Added or Changed](#)
- [Bugs Fixed](#)

### Functionality Added or Changed

- **Important Change:** New rows (`start_time_utc`, `end_time_utc`, `consistency_time_utc`, and `meb_version`) are being added to the `backup_history` table. To effect this change, the ALTER privilege on the `backup_history` table must be granted to the user by which `mysqlbackup` connects to MySQL Server before performing any backup operations with release 3.12.4.
- Information on the executed GTIDs is now included in the `mysqlbackup` output and the backup log when the backed-up server has GTIDs enabled. (Bug #25978803)
- A backup became corrupted if, during the backup process, a DDL operation that took advantage of MySQL server's `online DDL feature` occurred. This was because `mysqlbackup` did not support the server feature—and it still does not. This fix avoids the error by having `mysqlbackup` turn the server's system variable `old_alter_table` to “1” at the beginning of a backup if it is “0,” so that any DDL operations that take place during the backup are handled with the old table copy method. `mysqlbackup` then turns the variable back to “0” near the end of the backup operation.



#### Important

Notice that in cases where `mysqlbackup` quits unexpectedly and does not turn `old_alter_table` back to its original value, the user will have to turn the value back to “0” manually on the server, in order to return the server to its original configuration. This should be performed if the statement “Server system variable 'old\_alter\_table' was set to '0'. Setting it to '1'” appears in the early output of `mysqlbackup`, but the statement “Setting server system variable 'old\_alter\_table' back to '0'” does not appear before `mysqlbackup` quits.

(Bug #25217215)

- A new option, `--skip-final-rescan`, makes `mysqlbackup` skip the final rescan for InnoDB tables that are modified by DDL operations after the database has been locked near the end of a backup operation. This potentially shortens the duration for the lock and reduces the backup's impact on the server's normal operation. See the description for `--skip-final-rescan` for details. (Bug #21094221)
- The output by `mysqlbackup`, which goes to the `stderr` stream and the message log, has now been improved to include the timestamps and thread IDs for all steps taken by `mysqlbackup`, in order to provide more information for debugging purposes. (Bug #20142619)

- When there were no tables matching the regular expression specified with the `--include-tables` option during a backup operation, `mysqlbackup` still created a backup, which contained an empty folder for each database on the server. `mysqlbackup` now throws an error when `--include-tables` selects no tables to be backed up. (Bug #18114353)
- During the final stage of a backup when MySQL Enterprise Backup tried to temporarily put the database into a read-only state using the `FLUSH TABLES WITH READ LOCK` statement in order to copy non-InnoDB files, if a long query was running on the server at the same time, the `FLUSH TABLES WITH READ LOCK` statement could be taking too long to finish, holding up further queries and eventually bringing down the server.

A new `mysqlbackup` option `--lock-wait-timeout` can now be used to specify the timeout in seconds for the `FLUSH TABLES WITH READ LOCK` statement. If the timeout is exceeded, the statement is failed and the lock on the tables is released, so that queries held up by the lock can then be executed. `mysqlbackup` then retries the statement and continues with the backup. Default value for `--lock-wait-timeout` is 60 [seconds]. (Bug #14339483)

- In order to minimize the impact of a hot backup on the MySQL server, the copying of the buffer pool dump files and some of the metadata files is now performed before the final phase of the backup in which the server instance is locked. This shortens the duration for the lock and reduces the backup's impact on the server's normal operation.

Also, to minimize the resource used on a backup, the copying of the buffer pool dump files is no longer performed for partial and offline backups, for which the buffer pool dump is usually not very useful. (WL #9309, WL #9313)

## Bugs Fixed

- While MySQL Server interprets the system variable setting `--innodb_checksum_algorithm=0` to mean `--innodb_checksum_algorithm=crc32`, a `mysqlbackup` operation (except for backup) failed when `--innodb_checksum_algorithm=0` was set as a configuration option on the backed up server. With this fix, `mysqlbackup` now takes `--innodb_checksum_algorithm=0` as valid and interprets it as `--innodb_checksum_algorithm=crc32`. (Bug #28295519)
- On macOS, `mysqlbackup` failed to determine the relay log file name correctly if the relay log index file name was not specified. (Bug #25574605)
- The `Release` number for the RPM packages of MySQL Enterprise Backup always said "1," instead of giving the release number of the Linux distribution for which the package was built. (Bug #25538798)
- When the option `--only-innodb-with-frm` or `--no-locking` was used during a backup operation, the backup sometimes failed with `mysqlbackup` complaining that the highest LSN was larger in a copied page than on the backed-up server. It was because `mysqlbackup` did not perform a log flushing before copying the redo log when either of the mentioned options was used. With this fix, log flushing was always performed to prevent the error. (Bug #25412655)
- When creating a full image backup with the `--no-locking` option, `mysqlbackup` failed to write the binary log information to the backup history table and the `backup_variables.txt` file. The result was that when creating an incremental image backup based on the full backup, the attempt to copy the binary log files from the server into the incremental image backup (which is the default behavior) would fail, causing the incremental backup to stop. With this fix, the binary log information is no longer missing after the full backup, so the incremental image backup no longer fails." (Bug #25296324)

References: See also: Bug #19915713.

- After restoring an incremental backup of a slave server, the file `ibbackup_slave_info` was missing from the `meta` folder under the backup directory on the target server. (Bug #25097753)

- `mysqlbackup` logged double entries with wrong information into the `backup_history` table for optimistic backups. (Bug #24502876)
- Backups for a slave server in a multi-source replication setup using the `--slave-info` option failed. (Bug #24444950)

References: See also: Bug #21830316.

- Attempts to restore an image backup from the cloud using the `--skip-binlog` option failed with a “global tail magic mismatch” error. This was because `mysqlbackup` failed to perform a non-sequential read from the cloud with gaps caused by the skipping of the binary logs. This fix makes sure `mysqlbackup` can perform such reads. (Bug #23534700)
- An `extract` operation for an image backup failed with a checksum mismatch error in cases when, during the backup, an InnoDB tablespace file kept growing in size, and `mysqlbackup` failed to put the correct file size in its file header. (Bug #22905984)

References: This issue is a regression of: Bug #22613568.

- After an `apply-incremental-backup` operation on a full backup, `mysqlbackup` printed to the output stream and the message log file the old instead of the updated binary log position. (Bug #22085034, Bug #78898)
- When running the `backup-and-apply-log` command without a connection to the MySQL server, `mysqlbackup` could not know the correct binary log file name and binary log position for the backup; yet, at the end of the `backup-and-apply-log` operation, `mysqlbackup` still printed out some values for the binary log file name and position, which were random in nature. (Bug #21623089)
- A backup failed at the step when `mysqlbackup` applied the `FLUSH TABLES tbl_name [, tbl_name] ... WITH READ LOCK` statement on all non-InnoDB tables if any table names contained reserved words or special characters. It was because `mysqlbackup` did not enclose table names in backticks when issuing the statement, and this fix makes sure that is done. (Bug #19709505, Bug #74144)
- Occasionally, some files were missing from an image backup created by the `--backup-to-image` command. It was due to an internal race condition, which this fix eliminates. (Bug #19600687)

## Changes in MySQL Enterprise Backup 3.12.3 (2016-05-09)

- [Functionality Added or Changed](#)
- [Bugs Fixed](#)

### Functionality Added or Changed

- There are two enhancements on how you can configure SSL host authentication for cloud operations of MySQL Enterprise Backup:
  - A CA certificate directory, in addition to the default folder of the system, can now be specified with the new `--cloud-ca-path` option.
  - `mysqlbackup` now supports authentication using a CA bundle file, whose path is specified by the new `--cloud-ca-info` option.

See descriptions for the two new options in [Cloud Storage Options](#) for more information. (Bug #22761313)

- `mysqlbackup` used to sync all data from the buffer cache to the hard disk before closing all tables at the end of a backup operation. However, for systems with slow storage devices and databases with a huge number of tables, the sync would increase the backup time significantly. To shorten the backup time for those and other cases, starting with this release, the sync is no longer performed automatically. Users who want the sync to be performed at the end of a backup have to use the new `--free-os-buffers` option. (Bug #22561345)
- To avoid completing a backup of a slave server when temporary tables are still open on the slave, which will cause the restored slave server to be in an inconsistent replication state, `mysqlbackup` now has a new mechanism for ensuring that all temporary tables have already been closed before finishing a slave backup. See [Temporary tables on statement-based replication \(SBR\) replica](#) for details. A new option, `--safe-slave-backup-timeout`, has been created for specifying the time `mysqlbackup` will wait for all temporary tables to be closed before it times out. (Bug #19158516)
- The [compression options](#) can now be used with the `backup-and-apply-log` operation to create a directory backup that is prepared and compressed; the backup can then be restored using the `copy-back` operation and the `--uncompress` option. (Bug #18913565)
- During a `copy-back-and-apply-log` or a `copy-back` operation, `mysqlbackup` now checks that the specified values for the `innodb_log_files_in_group` and `innodb_log_file_size` options match those recorded in the backup's `backup-my.cnf` file, and throws an error if the values do not match. This prevents `mysqlbackup` from restoring the backup with the wrong parameters, which would result in a restored server that cannot be started. (Bug #14751027)

## Bugs Fixed

- An incremental or compressed backup might fail with an end-of-file error if there are large data files that kept growing during the backup. It was because, as the data files expanded, the write process altered the file sizes, which confused the read process for the same files. With this fix, file sizes and information on them is now properly handled. (Bug #23048004)

References: This issue is a regression of: Bug #19149210.

- Restoring a cloud backup sometimes failed with `Error 18: Transferred a partial file`. It was because `mysqlbackup` created wrong range headers for its REST requests for partial downloads. (Bug #23035334)
- `mysqlbackup` crashed when a `validate` operation was performed on an incremental backup that contained an undo tablespace but not a system tablespace. It was because `mysqlbackup` did not handle data files for undo tablespaces properly, and this fix corrects that. (Bug #22960185)
- Offline backups sometimes failed, with occasional crashes of `mysqlbackup`. (Bug #22595461)
- For a backup of a slave server, the file name of the master server's binary log and the binary log position for starting replication, which were stored in the file `backup_variables.txt` in the backup as `masterlog_file` and `masterlog_pos`, got corrupted when an `apply-log` or `copy-back-and-apply-log` operation was applied to the backup. (Bug #22329306)
- `validate` operations for backup images and `backup-to-image` operations left over a temporary folder (`/tmp`) after the operations were over. (Bug #20912357)
- Backups failed for a server that had once been started with the `--log-bin` option and then restarted without it. It was because `mysqlbackup`, seeing the old binary log index file on the server, looked in vain for the current binary log files, reported that they could not be found, and then exited. With this fix, `mysqlbackup` checks if binary logging is enabled for the server; if it is not, `mysqlbackup` then skips the copying of the binary log into the backup. (Bug #20873010)



## Changes in MySQL Enterprise Backup 3.12.2 (2016-01-21)

- [Functionality Added or Changed](#)
- [Bugs Fixed](#)

### Functionality Added or Changed

- If the value for the `innodb_checksum_algorithm` or `backup_innodb_checksum_algorithm` option provided on the `mysqlbackup` command line differed from that of the server, `mysqlbackup` gave a single warning that the checksum algorithm specified might be incompatible with the server. Starting with 3.12.2, to be consistent with the way checksums are handled by the MySQL server since 5.6.25, `mysqlbackup` gives a separate warning for every single page of data in every `.ibd` file for which the checksum algorithm specified does not match the one used on the server. (Bug #22509993)
- Values for `MASTER_USER` and `MASTER_PORT` are now included in the `CHANGE MASTER TO` statement in the slave information file (`meta/ibbackup_slave_info`) when the `--slave-info` option is used for backing up a slave server. (Bug #14213115)

### Bugs Fixed

- When `mysqlbackup` came across a file of an unknown file type and its path name contained characters that `mysqlbackup` could not convert to the file system character set, it threw an error. With this fix, `mysqlbackup` continues with its operation in the situation after giving a warning. (Bug #22098742)
- A backup failed if, towards the end of the backup process, `mysqlbackup` found the binary log file that was current at the beginning of the backup had been purged. With this fix, `mysqlbackup` now ignores the fact that the file has been purged, resets the log position to the now current binary log file, and continues with the backup without raising any issues. (Bug #21655145)
- During a backup, `mysqlbackup` performed, by default, an SQL query to get storage engine information that was to be put into the `backup_history` table. Because the query caused all table files on the server to be scanned, it consumed a great amount of IO resources when there were many tables on the server, resulting sometimes in serious performance issues. With this fix, only tables included in the backup are scanned, thus reducing the IO stress on the server. (Bug #21098174)
- When creating a compressed backup, `mysqlbackup` threw an error if a table on the server was dropped in the middle of the process. With this fix, the dropped table is ignored (as it does not need to be restored) and `mysqlbackup` finishes without throwing an error. (Bug #21087079)

References: See also: Bug #18358912.

- When a backup took a long time to perform and the binary logs were rotated in the middle of the process, `mysqlbackup` lost track of the binary log files it was copying, skipping the second last log file and attempting to copy the last one twice; that resulted in a file creation error, at which point `mysqlbackup` exited without releasing its lock on the tables in the database. With this fix, all binary log files are now copied properly, and the lock on the tables is released at the end of the backup process as usual. (Bug #20971763)
- When restoring an incremental backup image, if the binary log in the backup was larger than 16MB, the restored binary log would become corrupted, as `mysqlbackup` kept overwriting the same 16-MB file again and again with binary log contents. With this fix, the binary log is now correctly restored and has the same size as it did on the backed-up server. (Bug #20915642)
- A backup of a slave server failed if, during the backup, a relay log file got purged from the slave server (for example, due to a log file rotation). With this fix, backup continues even if `mysqlbackup` finds a relay log file missing. (Bug #20769891, Bug #76312, Bug #21655314, Bug #19255925)

- When the `--password` option was used without an argument with the `copy-back-and-apply-log` command, `mysqlbackup` did not prompt user for a password, but either took the password from the defaults files, or took it to be an empty string when no value was specified in the defaults files. (Bug #20657939)
- When the `trace` level of `mysqlbackup` messages was greater than “0,” if the operation command for `mysqlbackup` was invalid or missing, a stack trace and some error messages were printed, which made it look like `mysqlbackup` had crashed. With this fix, a new message is now shown before the stack trace, to better explain the situation. (Bug #20281022)
- If an incremental backup had already been applied to a directory backup with the `apply-incremental-backup` command and the up-to-date backup was then restored to a data directory, it was possible to restore the same incremental backup again to the data directory using the `copy-back-and-apply-log` command, potentially causing data inconsistencies. With this fix, the incremental data can be reapplied only when the `--force` option is used. Without the `--force` option, the `copy-back-and-apply-log` command skips the apply log operation if the incremental backup is a directory backup and throws an error if it is an image backup. (Bug #18004179)

## Changes in MySQL Enterprise Backup 3.12.1 (2015-06-29)

- [Security Notes](#)
- [Bugs Fixed](#)

### Security Notes

- MySQL Enterprise Backup has been updated to use OpenSSL version 1.0.1m, which has been publicly reported as not vulnerable to [CVE-2015-0286](#). (Bug #20747721)

### Bugs Fixed

- An `apply-log` operation on a backup created using transportable tablespace (that is, with the `--use-tts` option) failed if the backed-up database was using non-default page size for its InnoDB tables. (Bug #21030456)
- An incremental partial backup failed if the `--incremental-base` option was used. (Bug #20868507)
- When creating an incremental backup on a Windows platform using the `--incremental-base=dir:directory_path` option, if the `end_lsn` value for the last backup was  $2^{32}$  or larger, the value was retrieved as  $2^{32}-1$  by `mysqlbackup`. This was because `mysqlbackup` converted the value from string into long integer, instead of long long integer. This fix corrects the data type used during the conversion. (Bug #20839829)
- A `copy-back-and-apply-log` operation failed for an image backup if the value used for the `innodb_data_file_path` option during restore differed from the value got earlier from the server and stored in the `backup-my.cnf` file during backup. This was because the copy-back and the apply-log phases used the different values supplied for the option. With this fix, the value supplied during restore is used consistently. (Bug #20512332)
- A backup failed with the error message “Failed trying to flush for export” if the names of any table `mysqlbackup` tried to back up contained special characters (spaces, dashes, periods, dollar signs, etc.). With this fix, table names with special characters are now supported. (Bug #20454754, Bug #75722)
- A `validate` operation always failed for a compressed image backup with the complaint that the data file fragments were corrupted when the compressed image was big enough to be broken down into



multiple fragments. This was because `mysqlbackup` failed to recognize the file name extensions of the compressed fragments, and that has been corrected by this fix. (Bug #19796216)

- When the `--include-tables` or `--exclude-tables` option was provided with a regular expression for table names containing special characters, `mysqlbackup` was handling the special characters incorrectly, and hence table names were not filtered properly and tables were copied incorrectly. With this fix, `mysqlbackup` handles and filters the table names properly and takes backups of the matching tables. (Bug #19765644, Bug #11762160)

## Changes in MySQL Enterprise Backup 3.12.0 (2015-03-16)

- [Functionality Added or Changed](#)
- [Bugs Fixed](#)

### Functionality Added or Changed

- The `--skip-binlog` and `--skip-relaylog` options can now be used to skip the copying back of the binary log and relay log onto the server during a restore. This is particularly useful for users who do not want those logs appearing in the restored server's data directory, as that is always the location to which `mysqlbackup` will restore them, regardless of their original locations on the backed-up server. (Bug #19887998)
- MySQL Enterprise Backup no longer writes to the `incremental_base_lsn` column in the `mysql.backup_history` table when creating an incremental backup, as the column is no longer used by `mysqlbackup`. Note that the column will eventually be dropped from the table in future releases of MySQL Enterprise Backup. (Bug #19548604)
- The `--force` option can now be used during the restore of a full backup to overwrite existing data in a nonempty target directory. See the description for the `--force` option for details. (Bug #19266491)
- The binary log file and relay log files (in the case of a slave server) are now compressed when they are being included in a compressed backup and decompressed during a restore. (Bug #19149210)
- The start and end LSNs reported after an incremental backup was finished could be confusing to the users if there had been no changes to the database since the last full backup. With this fix, `mysqlbackup` simply reports that “No new log record found” in the situation. (Bug #18399069)
- MySQL Enterprise Backup now supports table renaming when a single table is restored from a backup created using [transportable tablespace \(TTS\)](#). See the description for the `--rename` option for details. (WL #7996)
- MySQL Enterprise Backup now supports cloud backup and restore using OpenStack Object Storage (“Swift”) 1.0. Authentication can be handled either through Swift's own TempAuth authentication system or the OpenStack Identity Service (Keystone) 2.0. A number of new command options have been introduced to support OpenStack Object Storage; see [Cloud Storage Options](#), for details. (WL #8130)

### Bugs Fixed

- **Microsoft Windows:** On Windows platforms, when symbolic links were involved in the file path arguments for options like `--backup-dir` or `--backup-image`, `mysqlbackup` failed with an error. With this fix, `mysqlbackup` works in the situation as long as the arguments do not involve dangling symbolic links, in which case `mysqlbackup` throws an error instead of creating any directory or file under any symbolically-linked locations. (Bug #19608231)
- When a base backup did not include the binary log or relay log files, updating the base backup up with a subsequent incremental backup using the `apply-incremental-backup` command would fail, unless

the incremental backup was created with the `--skip-binlog` and `--skip-relaylog` options. With this fix, the copying of the binary log and relay log is automatically skipped during an incremental backup when its base backup does not include those logs. (Bug #20594802)

- Performing a `backup-and-apply-log` and then a `copy-back-and-apply-log` caused the redo log files to be skipped during the restore process and thus missing on the restored server. This was because in this non-typical sequence of operations, the `apply-log` step (by which the redo log is usually copied) in the `copy-back-and-apply-log` step was skipped, as an `apply-log` operation had already been performed by the `backup-and-apply-log` step. This fix makes sure the copying of the redo log files is not skipped in that situation. (Bug #20583014)
- Near the end of the second phase of an optimistic backup, `mysqlbackup` rescanned the tables that had already been copied in the first, optimistic phase of the backup, and tried to copy again any of those tables that had been modified since they were first copied; this attempt to overwrite the already-copied table by a recopying caused a file creation error. With this fix, any such table changes are rightfully ignored, as the changes have been recorded in the redo log already and will be taken care of later by the `apply-log` operation. (Bug #20583014)
- If an incremental image backup contained a large number of pages from a single table, the restore of the incremental backup using the `copy-back-and-apply-log` command might fail. (Bug #20492274)
- Extracting individual files from an image backup using the `extract` command and the `--src-entry` option resulted in segmentation faults, if the binary log files had been included into the backup. (Bug #20458681)
- A `copy-back-and-apply-log` operation failed for an image backup if the `--backup-innodb_*` options (for example, `--backup_innodb_data_home_dir`, `--backup_innodb_log_group_home_dir`, and `--backup_innodb_undo_directory`) were used during the operation. With this fix, these options are ignored by `mysqlbackup` during the operation. (Bug #20451354)
- After a `RESET SLAVE` statement was executed on a slave server, a subsequent backup for the slave server failed with an error, as `mysqlbackup` could not copy the relay log file from the server. This was because `mysqlbackup` could not detect the location of the current relay log after the slave reset, and the this fix makes sure `mysqlbackup` knows how to do that. (Bug #20180440, Bug #75074)
- During the restore of an offline image backup, the `master.info` and `relay-log.info` files were sometimes not copied into the data directory on the target server. (Bug #19973192)

References: This issue is a regression of: Bug #19883801.

- Offline backups failed unless the `--skip-binlog` option was used, because the default action of copying the binary log failed. With this fix, the binary log can now be successfully included in an offline backup. (Bug #19941735)

References: This issue is a regression of: Bug #19883801.

- When creating a full backup with the `--no-locking` option, `mysqlbackup` failed to write the binary log information to the backup history table and the `backup_variables.txt` file. The result was that when creating an incremental backup based on the full backup, the attempt to copy the binary log files from the server into the incremental backup (which has been the default behavior of MySQL Enterprise Backup since 3.11.0) would fail, causing the incremental backup to stop. With this fix, the binary log information is no longer missing after the full backup, so the incremental backup no longer fails." (Bug #19915713)
- If a binary log file on the server got removed as a backup was taking place, the backup failed, as `mysqlbackup` could not find the binary log file it wanted to copy into the backup. With the fix,

`mysqlbackup` continues to finish the backup operation even if some binary log files have been deleted, except for the case of an incremental backup (for which the backup is still going to fail).

Also, with this fix, only the binary log files listed in the binary log index file are copied back to a server during a restore, so that purged binary log files, even backed up, are not restored. (Bug #19849326)

- When `mysqlbackup` encountered a corrupted `.frm` file during a backup, it threw an error, tried to continue with the backup, and then hung eventually. With this fix, `mysqlbackup` just gives a warning message (“`WARNING: An error occurred while adding manifest information for backup`”), and then continues with the backup as usual. (Bug #19608231)
- When trying to restore a non-TTS backup to a running server, `mysqlbackup` overwrote the data on the server without giving any warning. This fix makes `mysqlbackup` terminate the restore of a non-TTS full backup with an error whenever it finds the target data directory to be nonempty, and then issue a message that the `--force` option should be used if the user wants the original data to be overwritten.



#### Note

For overwriting the data directory during the restore of a full backup using the `--force` option, users are recommended to use the `copy-back` command, preceded by an `apply-log` operation, instead of using the one-step `copy-back-and-apply-log` command.

(Bug #19266491)

- The binary log and relay log index files included in a backup pointed to the files' original locations on the backed-up server. That might prevent the restored server to be started properly. With this fix, the log files' paths in the index files are properly updated during a backup to point to the files' locations in the backup directory. (Bug #19255992)
- Compression information displayed after the second phase of a compressed, optimistic backup was only for the second phase of the backup. With this fix, the information now reflects the total compression performed for the whole process, including both the first and second phases. (Bug #19200562)
- The `list-image` operation on cloud backups failed. This was because the operation requires transfer of data without buffering, but `mysqlbackup` was transferring data in buffered mode by default. This fix makes `mysqlbackup` download data for the operation without buffering—which is possible as long as the cloud proxy supports HTTP range headers. (Bug #19162974)
- When restoring an incremental backup using the `copy-back-and-apply-log` subcommand on Windows platforms, the operation failed when long file paths were used in the command-line options. (Bug #18448617)

## Index

### Symbols

`--force`, 9  
`--innodb_checksum_algorithm`, 3  
`--no-locking`, 3, 9  
`--only-innodb-with-frm`, 3  
`--skip-binlog`, 9  
`.frm` files, 9

### A

`ANSI_QUOTES`, 3

apply-log, 5, 8, 9

## **B**

backticks, 3  
backup-to-image, 5  
backup\_history, 9  
backup\_history table, 3  
binary log, 3, 7, 9  
binary log index file, 9

## **C**

cloud, 9  
cloud backup, 5  
cloud SSL connections, 5  
cloud-proxy, 9  
compressed, 5  
compression, 5, 9  
copy-back-and-apply-log, 7, 8, 9

## **D**

DROP TABLE, 7

## **E**

eof, 5  
extract, 9

## **F**

filesystem character, 7  
FLSUH TABLES, 3  
FREAK, 8  
free-os-buffers, 5

## **G**

GTID, 3

## **I**

ibbackup\_slave\_info, 3  
Important Change, 3  
incremental, 5  
incremental backup, 3, 7, 8, 9  
incremental restore, 9  
incremental\_base\_lsn, 9  
inndo\_undo\_tablespace, 5

## **L**

list-image, 9  
log-bin, 5

## **M**

master.info, 9  
Microsoft Windows, 9

## O

- offline backup, 5
- online DDL, 3
- OpenSSL, 8
- openstack, 9
- optimistic backup, 9
- overwrite, 9

## P

- partial backup, 8
- password, 7
- performance, 5
- privileges, 3

## R

- relay log, 7, 9
- relay log index file, 9
- relay-log.info, 9
- rename, 5, 9
- replication, 5
- restore, 5

## S

- slave-info, 7
- special characters, 8
- sql\_mode, 3
- storage engine, 7
- symbolic links, 9

## T

- trace, 7

## V

- validate, 5

