MySQL Enterprise Backup 8.0 Release Notes

Abstract

This document lists the changes to the MySQL Enterprise Backup 8.0 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 8.0 documentation, see the MySQL Enterprise Backup User's Guide (Version 8.0.27).

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: 2021-10-22 (revision: 23567)

Table of Contents

Preface and Legal Notices

This document lists the changes to the MySQL Enterprise Backup 8.0 product, beginning with the most recent release.

Legal Notices

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 8.0.27 (2021-10-19, General Availability)

MySQL Enterprise Backup 8.0.27 is the latest release for MySQL Enterprise Backup. It only supports MySQL Server 8.0.27. For earlier versions of MySQL 8.0, use the MySQL Enterprise Backup version with the same version number as the server. For MySQL server 5.7, please use MySQL Enterprise Backup 4.1.

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- The `--verbose` option is now synonymous with `--trace=1`. (Bug #32964563)
- The `--debug` option now accepts more arguments for providing various kinds of debugging information. See the option description for `--debug` for details. (Bug #32964563)
- It is now possible to create an incremental backup with page tracking with a DDL operation taking place in parallel with the backup. Some limitations apply; see more details here. (Bug #30603944)

Bugs Fixed

- Backup and restore operations might fail if arguments for the `--datadir` or `--backup_dir` options contained symbolic links. (Bug #33134292)
- During an incremental backup operation, `mysqlbackup` quit unexpectedly after receiving a segmentation error when the disk was full. This fix adds proper handling for a disk full error. (Bug #33123802)
- An `extract` operation for a compressed backup sometimes failed if an undo log truncation had taken place during the time the backup was created. (Bug #33082132)
- A backup operation failed when backing up tables that used the `keyring_hashicorp` plugin for InnoDB table encryption. It was because the HashiCorp Vault AppRole authentication secret ID supplied using the `--encrypt-password` option was obfuscated when sent to the server, and the issue has been fixed by this patch. (Bug #33020080)

  References: This issue is a regression of: Bug #32284801.
- A restore operation failed when, during the creation of the backup, some operation (for example, renaming) was performed on an external general tablespace. It was because `mysqlbackup` was

copying the tablespace twice to the same location on the target server, and that has been corrected by this fix. (Bug #32934170)

Changes in MySQL Enterprise Backup 8.0.26 (2021-07-20, General Availability)

MySQL Enterprise Backup 8.0.26 is the latest release for MySQL Enterprise Backup. It only supports MySQL Server 8.0.26. For earlier versions of MySQL 8.0, use the MySQL Enterprise Backup version with the same version number as the server. For MySQL server 5.7, please use MySQL Enterprise Backup 4.1.

- Packaging Notes
- Functionality Added or Changed
- Bugs Fixed

Packaging Notes

- Binary packages that include `curl` rather than linking to the system `curl` library have been upgraded to use `curl` 7.77.0.

Functionality Added or Changed

- Unnecessary single-table locks have been removed from backup operations. This allows performance improvement for backups that include many small tables. (Bug #32866300)
- The `--free-os-buffers` option can now take an argument, which tells `mysqlbackup` how to utilize `posix_fadvise()` (on supported platforms) for reducing the impact of `mysqlbackup` on the system performance. See the option description of `--free-os-buffers` for details. (Bug #32856108)
- Performance for backup operations that take place in parallel with DDL operations happening on the server is now improved by removing some unnecessary steps in the operations. (Bug #32855050)
- `mysqlbackup` now supports the keyring components on MySQL Servers: InnoDB tables encrypted by utilizing the `component_keyring_file` and `component_keyring_encrypted_file` can now be backed up and restored by `mysqlbackup`. See Working with Encrypted InnoDB Tablespaces for details.

Bugs Fixed

- `mysqlbackup` quit unexpectedly during a backup operation if the server had enabled redo and undo log encryption and the system variable `--innodb-page-size` had a non-default value. (Bug #32845253)
- The name of a log file created by `mysqlbackup` should be in the format of `MEB_`*`timestamp_operation`*`.log`, but the operation was missing in the filename. (Bug #32798284)
- During a backup operation involving encrypted InnoDB tables, the size of the decrypted InnoDB tablespace key file was not included in the log message for the decryption. (Bug #32787420)
- After upgrading from MySQL Enterprise Backup 8.0.18 or earlier, performance degradation was observed for large backup jobs involving more than 20,000 tables. That was caused by a number of different issues, which have been corrected by this fix. (Bug #32768465)
- An incremental backup using page tracking left a temporary file on the server's data directory. With this fix, the file is deleted at the end of the operation. (Bug #32666435)
- An incremental backup using page tracking failed if the server had a multi-file system tablespace that contained changed pages. (Bug #32580968)

- There were sporadic failures of backups involving encrypted InnoDB tables with the error "The encrypted table . . . is not initialized." This was caused by deferred flushing of the tablespace headers. This patch fixes the issue by enforcing timely flushing of the tables. (Bug #32495740)
- After an encrypted InnoDB table was restored from a TTS backup, a subsequent backup operation that included the restored table failed with the complaint that the table seemed corrupted. (Bug #32250682)
- A TTS backup took a long time to finish, even if it involved only a small amount of data. This was because `mysqlbackup` did not query for table information in an efficient way, and this patch corrects the issue. (Bug #30481117)

Changes in MySQL Enterprise Backup 8.0.25 (2021-05-11, General Availability)

MySQL Enterprise Backup 8.0.25 is the latest release for MySQL Enterprise Backup. It only supports MySQL Server 8.0.25. For earlier versions of MySQL 8.0, use the MySQL Enterprise Backup version with the same version number as the server. For MySQL server 5.7, please use MySQL Enterprise Backup 4.1.

This release contains no functional changes and is published to align the version number with the MySQL Server 8.0.25 release.

Packaging Notes

- Binary packages that include `curl` rather than linking to the system `curl` library have been upgraded to use `curl` 7.76.0.

Changes in MySQL Enterprise Backup 8.0.24 (2021-04-20, General Availability)

MySQL Enterprise Backup 8.0.24 is the latest release for MySQL Enterprise Backup. It only supports MySQL Server 8.0.24. For earlier versions of MySQL 8.0, use the MySQL Enterprise Backup version with the same version number as the server. For MySQL server 5.7, please use MySQL Enterprise Backup 4.1.

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- The `--rename` option now works with both full and partial restores:
  - If the `--include-tables` and `--exclude-tables` options are not used, all tables in the backup are restored, with the table selected by the `--rename` option renamed as specified.
  - If the `--include-tables` and `--exclude-tables` options are used, all tables selected by the two options together are restored, with the table selected by the `--rename` option renamed as specified.

  (Bug #31947026)

Bugs Fixed

- There were sporadic failures of backups involving encrypted InnoDB tables with the error "The encrypted table . . . is not initialized" when repeated DDL operations on the tables took place during the backup operations. (Bug #32495740)

- When backing up tables that used MySQL Enterprise Transparent Data Encryption (TDE), if no password was provided using the `--encrypt-password` option, `mysqlbackup` quit unexpectedly. With this fix, `mysqlbackup` throws a proper error and aborts the operation in the situation. (Bug #32486697)
- Backups failed when backing up tables that used the `keyring_hashicorp` plugin for InnoDB table encryption. (Bug #32284801)
- `mysqlbackup` quit unexpectedly at the end of a backup to a tape using the NetWorker software. (Bug #32081275)
- The `--include-tables` and `--exclude-tables` options were ignored during a restore of a TTS backup taken with full locking (i.e., with `--use-tts=with-full-locking`). (Bug #31947026)
- A restore of a compressed backup failed if the backed-up server did not use `.ibd` as the extension for InnoDB tablespaces. (Bug #31596356)
- Backups for a server containing encrypted InnoDB tables failed when the server was of 64-bit and `mysqlbackup` was of 32-bit (or vice versa). It was due to the way keyring files were handled in the situation, which has been fixed by this patch. (Bug #29292085)

Changes in MySQL Enterprise Backup 8.0.23 (2020-01-18, General Availability)

MySQL Enterprise Backup 8.0.23 is the latest release for MySQL Enterprise Backup. It only supports MySQL Server 8.0.23. For earlier versions of MySQL 8.0, use the MySQL Enterprise Backup version with the same version number as the server. For MySQL server 5.7, please use MySQL Enterprise Backup 4.1, and for MySQL Server 5.6, please use MySQL Enterprise Backup 3.12.

Functionality Added or Changed

- Logging on cloud operations with OCI object storage now provides more information. (Bug #32011770)
- For a cloud backup operation to an Amazon S3-compatible storage service, a check on whether the bucket exists on the storage service has been added to the beginning of the operation. If the specified bucket does not exist, `mysqlbackup` throws an error and quits the operation. (Bug #31981595)
- A new option, `--cloud-chunk-size`, has been introduced for specifying the size of a chunk when chunked transfer is enabled for cloud operations. See the description for `--cloud-chunk-size` for details. (Bug #31977600)
- MySQL Enterprise Backup has extended the types of cloud storage services it supports; see Cloud Storage Options for details.

Bugs Fixed

- A restore or validate operation for a compressed backup failed with an `unexpected end of file` error. (Bug #32163271)
- `.sdi` files were not included in partial backups, even when they were matched by the regular expression given in the `--include-tables` option. (Bug #32162426)

- When an incremental backup was created with redo log only and the redo log portion in it did not contain the latest InnoDB checkpoint of the backed-up server, after the incremental backup was restored and the server was restarted, the server reported that the data was corrupted. It was because the restore process replaced the redo log files already restored onto the server with the redo log data from the incremental backup, causing the latest checkpoint already on the server to be lost. With this fix, when an incremental backup created with redo log only was restored, the redo log files already on the server was only appended but never replaced, in order to avoid the loss of the latest InnoDB checkpoint that has been restored. (Bug #32139949)

- When a password was not specified with the `--encrypt-password` option for a `validate` operation, `mysqlbackup` threw an error, complaining that the password was missing. With this fix, `mysqlbackup` prompts for the password under the situation. (Bug #32037428)

- `mysqlbackup` quit unexpectedly during a restore operation if the `--datadir` option was not used in the restore command. With this fix, `mysqlbackup` throws an error and quits gracefully under the same situation. (Bug #31947239)

- A backup using redo log archiving failed, because `mysqlbackup` created a subdirectory under the redo log archiving directory (specified in `innodb_redo_log_archive_dirs`) that was accessible to all OS users, and that was not allowed. With this fix, a subdirectory with the proper permissions is created under the situation. (Bug #31926082, Bug #100913)

- On a Windows platform, when the value of the server's system variable `innodb_redo_log_archive_dirs` was an absolute path name without a label, a backup operation for the server using redo log archiving failed with an internal error. This was due to `mysqlbackup` misinterpreting the path name in the situation. With this fix, the path name is now properly interpreted, and if the archive does not exist, a proper error is thrown. (Bug #31900686)

- Cloud backups to OpenStack Swift or compatible object storage services using HTTP basic authentication failed with an HTTP Error 411. (Bug #31847208)

- After a TTS backup containing partitioned tables and encrypted InnoDB tables was restored, a `DROP DATABASE` statement on the restored server failed. (Bug #31847208)

- An incremental backup taken with redo log only was completed by `mysqlbackup` without throwing an error even if an in-place DDL had taken place during the incremental backup. This would cause an assertion error when the server was restarted after the incremental backup was restored. With this fix, `mysqlbackup` throws an error during an incremental backup when an in-place DDL takes place. (Bug #31653902)

- An `image-to-backup-dir` operation on a cloud backup failed with the complaint by `mysqlbackup` that it was not a supported cloud operation, even though the `extract` command, an alias of `image-to-backup-dir`, worked. (Bug #31453397)

Changes in MySQL Enterprise Backup 8.0.22 (2020-10-19, General Availability)

MySQL Enterprise Backup 8.0.22 is the latest release for MySQL Enterprise Backup. It only supports MySQL Server 8.0.22. For earlier versions of MySQL 8.0, use the MySQL Enterprise Backup version with the same version number as the server. For MySQL server 5.7, please use MySQL Enterprise Backup 4.1, and for MySQL Server 5.6, please use MySQL Enterprise Backup 3.12.

- Functionality Added or Changed

- Bugs Fixed

Functionality Added or Changed

- MySQL Enterprise Backup now supports cloud backup and restore using the Object Storage service of Oracle Cloud Infrastructure (OCI) with Pre-Authenticated Request (PAR) URLs. A new option, `--cloud-par-url`, has been introduced for the purpose. See Backing Up to Cloud Storage and Restoring a Backup from Cloud Storage to a MySQL Server for details.

  Also, OAuth is no longer supported by MySQL Enterprise Backup for authentication with the OCI Object Storage service. (Bug #31921624)

- Messages printed by `mysqlbackup` were sometimes truncated, because of the fixed message length. The length of messages is now extensible, so messages are no longer truncated. (Bug #31433762)

- MySQL Enterprise Backup now supports S3-compatible cloud storage services with a new option `--cloud-host`, by which users can specify the hostname of the storage service.

- MySQL Enterprise Backup now supports user authentication by the server using LDAP. Two new options, `--plugin-dir` and `--enable-cleartext-plugin`, have been introduced to support this feature. See Using LDAP for Server Authentication for details.

Bugs Fixed

- When the `keyring_file` and `keyring_udf` plugins were enabled on a server and a new key was generated, a subsequent backup failed. It was because `mysqlbackup` could not copy the generated key, and this patch corrects the issue. (Bug #31717154)

- During a backup, if redo log archiving was disabled on the server, `mysqlbackup` printed the message "Failed to start redo log archiving." With this fix, `mysqlbackup` prints instead the more appropriate message "Redo Log Archiving is disabled." (Bug #31618079)

- An optimistic backup failed if an undo log truncation took place during the backup. (Bug #31544201)

- A restore of a compressed backup failed if an undo log truncation took place on the server during the backup process. (Bug #31544087, Bug #31544087)

- A backup failed with a segmentation fault when the server to be backed up was once a member of a Group Replication setup that had been dissolved. It was because `mysqlbackup` still attempted to treat the server as a Group Replication node, and this patch corrects that. (Bug #31507782)

- When backing up a server that has been upgraded from MySQL 5.7 to 8.0, `mysqlbackup` copied only the first file for the InnoDB system tablespace into the backup when there were more of them to be copied. (Bug #31485318)

- A backup failed with `mysqlbackup` quitting with a segmentation fault if the server had redo log archiving enabled and then a query for the server status variable `Innodb_redo_log_enabled` failed. (Bug #31483606)

- In the `backup_create.xml` file inside the meta folder of a backup, the saved options inside the <server_repo_opts> section had the `backup_` suffix added to their option names by mistake. This fix removes the suffix for those options. (Bug #31459476)

  References: This issue is a regression of: Bug #31370902.

- During any `extract` operation, `mysqlbackup` printed the message "`Failed to get file status with error no : 1504`." As the message might mislead users to think there was an error, it has now been removed from the output for trace level `INFO` . (Bug #31453496)

- When a `.idb` file of an encrypted partitioned table disappeared during a backup operation, `mysqlbackup` kept complaining for 500 seconds that the file had vanished until it threw an error. With this fix, `mysqlbackup` threw an error immediately when running into the situation. (Bug #31451654)
- `mysqlbackup` hung when, during a backup, a parallel DDL occurred and some commits on the server failed. It was because in the situation, `mysqlbackup` could not obtain a backup lock on the server. With this fix, the attempt to obtain the backup lock times out after a while, and `mysqlbackup` continues with the backup. (Bug #31450946)
- During a backup, if `mysqlbackup` failed to query from the server the undo log information for some reasons, the backup failed with a segmentation error. With this fix, `mysqlbackup` proceeds with the rest of the backup process when running into the same situation. (Bug #31445229)
- During a backup operation, if an undo log tablespace was deleted right before the final rescan phase of the backup operation, `mysqlbackup` quit unexpectedly. It was due to the way `mysqlbackup` handled undo log information, which has been fixed by this patch. (Bug #31445229)
- During a backup operation, if `mysqlbackup` failed with its query for the server status variable `Innodb_buffer_pool_dump_status`, it ignored the error and then hung. (Bug #31445204)
- When backing up to a tape using Oracle Secure Backup (OSB) 12.2.0.1, `mysqlbackup` was disconnected from the server and backup ended with a segmentation fault. (Bug #31442335)
- A MySQL Enterprise Server quit unexpectedly when the `mysqlbackup` component was installed and uninstalled in two different client sessions, and the `mysqlbackup_page_track_set()` function was invoked from a third session. (Bug #31383239)
- A TTS backup failed if the server to be backed up had a full-text index and also a table named `fts`, and the table was matched by the regular expression for the partial backup. (Bug #31382819)
- A backup failed if tables were dropped from the database when the backup was running. It was due to the wrong timing for applying the backup lock by `mysqlbackup`, which has been fixed by this patch. (Bug #31331051)
- An incremental backup using page tracking sometimes resulted in a corrupted backup if DML operations took place on the server during the backup. Beyond fixing the issue, this patch also adds validation for data pages in a page-tracking incremental backup, and makes it possible to run page-tracking incremental backups for databases with a multiple-file system tablespace. (Bug #31329848)
- A compressed backup became corrupted when a DDL operation took place on the server during the backup process. (Bug #31321514)
- In a `backup-and-apply-log` operation, the connection to the server was kept open even after the backup phase of the operation was over. With this fix, the connection is closed when the operation enters the apply-log phase, in order to free up resources. (Bug #30012743)
- A backup failed when the server's system variable `--innodb-data-file-path` pointed to a raw partition on the hard disk, because in that case, the OS returned a zero device size for the raw partition to `mysqlbackup`. With this fix, `mysqlbackup` took the device size directly from the value of `--innodb-data-file-path`. (Bug #27811936)

Changes in MySQL Enterprise Backup 8.0.21 (2020-07-13, General Availability)

MySQL Enterprise Backup 8.0.21 is the latest release for MySQL Enterprise Backup. It only supports MySQL Server 8.0.21. For earlier versions of MySQL 8.0, use the MySQL Enterprise Backup version with

the same version number as the server. For MySQL server 5.7, please use MySQL Enterprise Backup 4.1, and for MySQL Server 5.6, please use MySQL Enterprise Backup 3.12.

In the documentation for MySQL 8.0.21, we have started changing the term “master” to “source”, the term “slave” to “replica”, the term “whitelist” to “allowlist”, and the term “blacklist” to “blocklist”. There are currently no changes to the product's syntax, so these terms are still present in the documentation where the current code requires their use. See the blog post MySQL Terminology Updates for more information.

Packaging Notes

- For Windows, MSI installer packages for MySQL Enterprise Backup now include a check for the required Visual Studio redistributable package, and produce a message asking the user to install it if it is missing. (Bug #30541398)

Functionality Added or Changed

- **Important Change:** The storage engine for the `mysql.backup_sbt_history` table on a backed-up server has switched from CSV to InnoDB. Also, an auto-increment primary key `id` column has been added to the table. When working with a Group Replication setup, `mysqlbackup` now makes the `backup_sbt_history` table available to all members of the server group by making sure that the table is updated on a primary node during each `mysqlbackup` operation.

  When MySQL Enterprise Backup 8.0.21 or later tries to perform its first full backup on a database using the SBT API (see Backing Up to Tape with Oracle Secure Backup for details), it automatically checks the format of the `mysql.backup_sbt_history` table. If it detects that the table is in the old format (which means the server has been upgraded from 8.0.20 or earlier and has been backed up by MySQL Enterprise Backup before using the SBT API), it attempts to perform an update on the table automatically. Grant these privileges, required for the table upgrade, to the `mysqlbackup` user on the server:

  ```
  GRANT ALTER ON mysql.backup_sbt_history TO 'mysqlbackup'@'localhost';
  GRANT CREATE, INSERT, DROP ON mysql.backup_sbt_history_old TO 'mysqlbackup'@'localhost';
  GRANT CREATE, INSERT, DROP, ALTER ON mysql.backup_sbt_history_new TO 'mysqlbackup'@'localhost';
  ```

  See SBT Backup History Table Update for details. (Bug #30537077)

- **Important Change:** For a `backup-to-image` operation, when a relative path is specified for the `--backup-image` option, `mysqlbackup` now interprets the file path given as relative to the backup directory.

  References: See also: Bug #30935456.

- Encrypted InnoDB tables can now be included in partial backups and restores using transportable tablespaces (TTS). (Bug #31796017)

- The `tool_name` column of the `backup_progress` table on the MySQL server is now populated with the full `mysqlbackup` command that invoked a backup operation. (Bug #31011043)

- The file `backup_gtid_executed.sql` was not included in a TTS backup for a replica server using GTIDs. The file is now included in a TTS backup as long as the `--slave-info` option is used. (Bug #30925447)

- A backup now fails when a binary or relay log file is purged while the backup is going on; it also fails when `mysqlbackup` finds a binary log file missing on the server (however, if a relay log file is missing, the backup continues). (Bug #29269039)
- Commands for operations on incremental backups (`copy-back`, `copy-back-and-apply-log`, `apply-log`) have been simplified: the `--incremental` option is no longer needed for those operations.
- Commands for operations on compressed backups (`copy-back`, `copy-back-and-apply-log`, `apply-log`, etc.) have been simplified: the `--uncompress` option is no longer needed, except for `extract` and `image-to-backup-dir` operations that *do not* use the `--src-entry` option.
- Compressed InnoDB files are now being verified in `validate`, `backup`, and `backup-to-image` operations.
- Encrypted InnoDB tables are now being verified in `validate` operations.

Bugs Fixed

- When a restore failed for a compressed backup because the `--compress-method` option was used, `mysqlbackup` did not print a meaningful error message. With this fix, the error message indicates that the option is incompatible with the operation. (Bug #31861826)
- When creating an image backup, if the backup directory (specified with `--backup-dir`) was full, the backup operation still finished, with just a warning. Trying to restore the backup then caused `mysqlbackup` to quit unexpectedly. With this fix, the backup fails with an error when the backup directory is full. (Bug #31370902)
- Backups might fail for a MySQL Server 8.0.20 that was upgraded from an earlier server version, with `mysqlbackup` complaining that the first system tablespace file (`ibdata1` usually) was corrupted. It was due to the way MySQL Server 8.0.20 handled the system tablespace, which `mysqlbackup` had not adapted itself to, causing an error sometimes with an upgraded server. This patch adjusted `mysqlbackup` to work properly with the server. (Bug #31263411)
- When the `--src-entry` option was used with the `list-image` command, `mysqlbackup` did not reject the option at once, but finished the command and then threw an `Invalid Argument` error. With the fix, `mysqlbackup` threw an `Incompatible Option` error immediately in the situation. (Bug #31255087)
- A restore operation for an incremental backup failed when the `--with-timestamp` option was used. (Bug #31184454)
- An `extract` operation failed with `mysqlbackup` complaining that there was no table match when the option `--src-entry` was set to `meta/backup_variables.txt`. With this fix, `mysqlbackup` no longer throws an eorrr in the situation, but prints the message “The src-entry 'backup_variables.txt' is by default extracted to the output directory”. (Bug #31180805)
- On non-Windows platforms, when the `--force` option was used with a table-level restore (a partial restore of selected tables) of a non-TTS backup, the redo log files on the server were deleted by `mysqlbackup`. (Bug #31173210)
- After a MySQL Server containing encrypted InnoDB tables was upgraded from series 5.7 to 8.0, backup operations on it failed with a Keyring Error. It was due to the way the keyring was handled by `mysqlbackup` in the situation, which has been fixed by this patch. (Bug #31137866)
- When the `--backup-image` option was used in a `backup` operation for a directory backup, `mysqlbackup` ignored the option and continued to perform a directory backup. With this fix, `mysqlbackup` throws an incompatible option error in the situation. (Bug #31137103)

- `mysqlbackup` returned an `Internal Error` when a compressed backup created with the `--use-tts=with-full-locking` option was being restored. (Bug #31061894)
- When backing up a replica server, if some relay log files were missing, the backup was still completed as expected, but `mysqlbackup` printed out error messages. With this fix, `mysqlbackup` returns success instead in the situation. (Bug #31059294)
- When the `--backup-image` option was used with the `backup-and-apply-log` command, `mysqlbackup` finished the command as usual, even though the option and the command are not compatible. With this fix, in the situation, `mysqlbackup`, gives a warning that the `--backup-image` option is ignored. (Bug #31001191)
- When a single-file backup was created with the `--with-timestamp` option and a relative path was specified for `--backup-image`, the image backup was created under the current working directory (which had been the expected behavior since release 8.0.19), but not in a subdirectory that bore the timestamp in its name.

  With this fix, the location for the backup in the situation has been changed: for a `backup-to-image` operation, the relative path given with `--backup-image` is now taken as relative to the backup directory, and if the `--with-timestamp` option is used, the backup is created under the backup directory in a subdirectory that bears the timestamp in its name. (Bug #30935456)
- When backing up to a tape through a media management software (MMS), `mysqlbackup` always set a default value of `0000-00-00 00:00:00` for the `file_creation_time` and `file_expiry_time` values for the operation's entry in the `backup_sbt_history` table on the backed-up server. If the backup failed for some reasons, those zero values were then written to the table. If, later, the `backup_sbt_history` table was queried in NO_ZERO_DATE or NO_ZERO_IN_DATE SQL mode, the server returned `ERROR 1194 (HY000): Table 'backup_sbt_history' is marked as crashed and should be repaired`. With this fix, in the case of a backup failure, `mysqlbackup` writes the current time during the backup to those values, so the time values will never be zeros. (Bug #30275637)
- When the `--skip-binlog` option was used with a restore operation of a TTS backup, the operation failed. With this fix, the option is ignored in the situation. (Bug #29813666)
- When the `--compress-method` option was set to `none`, the backup was finished without compression as expected, but `mysqlbackup` printed erroneous compression information and saved the InnoDB tablespace files with the `.ibz` extension. With this fix, the described behaviours of `mysqlbackup` no longer occur in the situation. (Bug #29806518)
- The `--compress-level` option took up a value of `0` instead of the default value of `1` when the `--compress-method` option was used without the `--compress` option. With this fix, the default value of the option is always honored (for the applicable compression methods). (Bug #29806518)
- A restore operation failed for a backup image created with the `backup-dir-to-image` command from a directory backup, if the backed-up server used a keyring plugin other than `keyring_encrypted_file` for InnoDB table encryption. It was because the `backup-dir-to-image` operation mishandled the `keyring_kef` file in the backup, and this patch corrects the problem. (Bug #27874581)

Changes in MySQL Enterprise Backup 8.0.20 (2020-04-27, General Availability)

MySQL Enterprise Backup 8.0.20 is the latest release for MySQL Enterprise Backup. It only supports MySQL Server 8.0.20. For earlier versions of MySQL 8.0, use the MySQL Enterprise Backup version with

the same version number as the server. For MySQL server 5.7, please use MySQL Enterprise Backup 4.1, and for MySQL Server 5.6 and 5.5, please use MySQL Enterprise Backup 3.12.

Functionality Added or Changed

- The tablespace_tracker file has been simplified: it now contains only two fields for each external tablespace: `server_file_path` and `space_id`. `mysqlbackup` no longer relies on the file for information on the `backup_file_path` and the tablespace type, which means that users no longer need to update the tablespace_tracker file when they move a directory backup to a new location.
- Table-Level Recovery (TLR) is a new feature of MySQL Enterprise Backup that allows selective restores of tables or schemas from full backups; see Table-Level Recovery (TLR) for details.
- The legacy option `--include` is now deprecated. A deprecation warning is now issued by `mysqlbackup` whenever the option is used. The `--include-tables` and `--exclude-tables` options should be used instead for partial backups and restores.

Bugs Fixed

- A backup failed with `ERROR: Bad table space file header` when the server had more than one system tablespace files. It was because `mysqlbackup` looked for the tablespace file header at the wrong place, and this patch corrects the problem. (Bug #30983009)
- During an incremental backup, `mysqlbackup` simply repeated the Server Repository Options when trying to print the Backup Configuration Options in its output. (Bug #30948251)
- Backups failed when the server used a keyring plugin and its `sql_mode` was set to `ANSI_QUOTES`. It was because `mysqlbackup` used the wrong kind of quotes in the situation when querying the server, and that has been fixed by this patch. (Bug #30920140)
- An incremental optimistic image backup failed when the server was started with a non-default `innodb_data_file_path` value containing more than one InnoDB system tablespace file. It was because `mysqlbackup` could not handle the situation in which the two different files had the same space ID, and this patch fixes the problem. (Bug #30914039)
- A partial restore of a TTS backup failed when the file path specified with the `--datadir` option contained extra slashes (/) when compared with the data directory file path the server was started with. With this fix, such extra slashes for the `--datadir` option are ignored. (Bug #30834688)
- `mysqlbackup` might hang when, during a backup, an `ALTER TABLE ... DROP PARTITION` statement was run in parallel on the server. It was because with the partition deleted, any subsequent DDL operation within any unclosed transaction would cause a DDL lock on the table involved, and `mysqlbackup` hung when it also wanted to lock the table for the backup operation. With this fix, the unclosed transaction is rolled back, so the lock due to the failed DDL statement is avoided, and `mysqlbackup` proceeds as usual with its backup. (Bug #30599785)
- During a backup operation, if any tablespace's encryption status was changed (for example, from encrypted to unencrypted or vice versa, and even if the table was eventually changed back to its original encryption status), `mysqlbackup` reported success, but it quit unexpectedly during the restore operation of the backup because of its inconsistency. With this fix, the encryption statuses of tablespaces are properly tracked throughout the backup operation, so that the tables are consistently backed up. (Bug #30599476)

- When the `--src-entry` option was used with the `extract` command, a trailing slash in its value (for example, in `foo/`) was ignored, so that instead of extracting from the backup only those directories whose names ended with the value (for example, `datadir/foo/`), `mysqlbackup` also extracted all files whose paths contained the value (for example, `datadir/bar/foo.sdi`). With this fix, the trailing slash is honored, and it only causes folders whose names end with the value to be extracted.

  It is also clarified in the documentation that the value of the `--src-entry` option is actually used to match any files or non-empty folders that contain the value in their names, and a trailing slash is interpreted as described in the last paragraph. (Bug #30461403)

- When there was a user-created `mysql.backup_progress` table on a server that was being backed up, `mysqlbackup` finished the backup successfully, but also printed error messages and recorded a backup failure in the `backup_history` table. With this fix, the backup is finished as normal with a warning. (Bug #30351172)

- The binary log basename appeared as an empty string in the progress report of a `copy-back-and-apply-log` operation. (Bug #29936558)

- When a data tablespace had the same name as an undo tablespace on the server, a compressed backup containing the tablespace could be created by `mysqlbackup`, but the backup could not be restored due to the filename conflict. With this fix, the backup fails in the situation. (Bug #29881640)

- A backup failed when it involved encrypted InnoDB tables and the `--skip-unused-pages` option was used. (Bug #29861298)

- When a compressed backup was created with the `backup-and-apply-log` command and then restored using the `copy-back-and-apply-log` command, the redo log were missing from the restored server, causing an InnoDB error when the server was started. (Bug #29851603)

- A backup failed when the `--skip-unused-pages` and `--optimistic-busy-tables` options were used together. (Bug #29840923)

- When the server to be backed up has `super_read_only=ON`, `mysqlbackup` gave the warning that the backup operation could not be logged even if the `--no-history-logging` option has already been used with the backup command. This patch removes the unnecessary warning. (Bug #29742011)

- A `backup-and-apply-log` operation failed for a TTS backup if the `--compress` option was used. (Bug #29639871)

- An `extract` operation for the file `meta/backup_variables.txt` failed with `mysqlbackup` complaining that the value of the option `--src-entry` did not match any table in the backup. With this fix, `mysqlbackup` no longer throws an eorrr in the situation, but prints the message “The src-entry 'backup_variables.txt' is by default extracted to backup-dir”. (Bug #29519710)

- A restore of a directory backup failed if there existed tablespaces outside of the data directory on the backed-up server and the directory backup has been renamed before the restore. (Bug #29265179)

- During a backup operation, `mysqlbackup` printed messages regarding the encryption keyring even though the server did not utilize InnoDB table encryption. With this patch, `mysqlbackup` stops printing such messages in the situation. (Bug #29151380)

Changes in MySQL Enterprise Backup 8.0.19 (2020-01-13, General Availability)

MySQL Enterprise Backup 8.0.19 is the latest release for MySQL Enterprise Backup. It only supports MySQL Server 8.0.19. For earlier versions of MySQL 8.0, use the MySQL Enterprise Backup version with

the same version number as the server. For MySQL server 5.7, please use MySQL Enterprise Backup 4.1, and for MySQL Server 5.6 and 5.5, please use MySQL Enterprise Backup 3.12.

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- **Important Change:** For the `backup-to-image`, `extract`, `list-image`, and `copy-back-and-apply-log` commands, any *relative path* specified with `--backup-image` is now taken to be relative to the current directory in which the command is run. (Bug #29943103)
- **Important Change:** The storage engine for the `mysql.backup_progress` table on a backed-up server has switched from CSV to InnoDB. Also, an auto-increment primary key `id` column plus a composite index on the table's `backup_id` and `current_timestamp` columns have been added. When working with a Group Replication setup, `mysqlbackup` now makes the `backup_progress` table available to all members of the server group by making sure that the table is updated on a primary node during each `mysqlbackup` operation.

  When MySQL Enterprise Backup 8.0.19 or later tries to perform its first full backup on a database, it automatically checks the format of the `mysql.backup_progress` table. If it detects that the table is in the old format (which means the server has been upgraded from 8.0.18 or earlier and has been backed up by MySQL Enterprise Backup before), it attempts to perform an update on the table automatically. Grant these privileges required for the table upgrade to the `mysqlbackup` user on the server:

  ```
  GRANT ALTER ON mysql.backup_progress TO 'mysqlbackup'@'localhost';
  GRANT CREATE, INSERT, DROP ON mysql.backup_progress_old TO 'mysqlbackup'@'localhost';
  GRANT CREATE, INSERT, DROP, ALTER ON mysql.backup_progress_new TO 'mysqlbackup'@'localhost';
  ```

  See Backup Progress Table Update for details. (Bug #29882544, Bug #28695518)
- **Important Change:** Due to the added protection for the backups against inconsistency caused by parallel DDL operations (see discussions on the new feature below), the user by which `mysqlbackup` connects to the MySQL server must now be granted the `SELECT ON *.*` privilege; see Grant MySQL Privileges to Backup Administrator.
- The logging for backup restore has been improved: at the steps for setting the sizes of the log files, the names of the log files are now included. (Bug #30380310)
- `mysqlbackup` now prints a stack trace after being terminated by a signal. (Bug #30042338)
- When `mysqlbackup` fails to connect to a server, the warning returned by `mysqlbackup` now includes the hostname and port number for TCP connections, and the socket information for socket connections. This is particularly helpful for a Group Replication setup, for which `mysqlbackup` might attempt to connect to more than one host. (Bug #30040027)
- If a binary log file could not be opened during a non-incremental backup, the backup would still be completed, but the `backup_history` table would indicate the backup had failed. With this fix, backups of all types fail if any relevant binary log files could not be opened, and a proper error is thrown. (Bug #29882381)
- `mysqlbackup` now includes the configuration files `auto.cnf` and `mysqld-auto.cnf` from a server in its backup (except for a TTS backup). They are restored to the target server's data directory as `backup-auto.cnf` and `backup-mysqld-auto.cnf` respectively. To use those files to configure your restored server, rename them to their original names before starting the server. (Bug #27121423, Bug #30033486)

- Master key rotation for binary log encryption on the server in between a full and an incremental backup, as well as between two incremental backups performed by `mysqlbackup`, is now supported. During an incremental backup, `mysqlbackup` now records encryption information for all the encrypted binary log files (including those already backed up in earlier full or incremental backups) unless the `--skip-binlog` option is used, in which case a warning is given that the older binary log files might become unrestorable.

  Also, the `--skip-binlog` option now makes binary log to be skipped not just for the current backup operation, but also for all subsequent incremental backups that are based on the current backup. And when an incremental backup is restored with the `--skip-binlog` option or when the incremental backup just does not contain the binary log, `mysqlbackup` renames any binary log and index files that have already been restored with the base backup by adding to them the `.old` extension, and then gives a warning.

- In the past, if DDL operations took place when a backup was in progress, the backup might become inconsistent. It is now safe to have DDL operations (CREATE TABLE, RENAME TABLE, DROP TABLE, ALTER TABLE, and operations that map to ALTER TABLE like CREATE INDEX) happening on the server in parallel with a backup operation as long as:

  - The tables involved exist in their own tablespaces, instead of being in the system tablespace or some general tablespaces.
  - These server features have not been applied to the tables involved:
    - Data-at-rest encryption
    - Page-level compression
    - Full-text indexing
  - The backup is not taken with the following `mysqlbackup` features:
    - Optimistic backup
    - Transportable tablespace (TTS)
    - Redo log archiving
    - Incremental backups with-redo-log-only
    - Incremental backup using page-tracking

Bugs Fixed

- When an image backup was created using the `backup-dir-to-image` command from a compressed directory backup containing InnoDB tables in the COMPRESSED row format, the image backup could not be validated, unless it was first unpacked back into a directory. (Bug #31346149)
- `mysqlbackup` reported that an `extract` operation succeeded even if `--src-entry` specified a file that did not exist in the backup. This fix adds a validation for the `--src-entry` value and makes `mysqlbackup` throw an error if the validation fails. (Bug #30461566)
- During an `extract` operation, the extracted file contents did not go into `stdout` as expected when `--dst-entry=-`, but into a file named “-” instead. (Bug #30451238)
- Sometimes, after receiving a signal 6 due to some errors, `mysqlbackup` quit without printing any error messages. (Bug #30423128)

- `mysqlbackup` sometimes hung during a backup when redo log archiving had been enabled. It was due to the way `mysqlbackup` switched between reading the redo log files and the redo log archive, which has been corrected by this fix. (Bug #30387689)

- If a tablespace on a server was deleted in the middle of a backup process and then restored before the end of the process so that the backup included the table and was successful, `mysqlbackup` still reported in the `mysql.backup_history` table that the backup failed. (Bug #30340161)

- At restart, a restored server someimtes gave the warning `Doublewrite page ### for {space: ###, page_no:###} could not be restored`. This happened because the doublewrite buffer, being restored from the backed up server, contained pages that were no longer relevant. With this fix, the doublewrite buffer in the backup was cleared during the backup process, so that it is no longer restored. (Bug #30286862)

- When OpenSSL 1.1.1 was used for connecting `mysqlbackup` to the server and the `--tls-version` option was not specified, TLSv1.3 was used, but the output of `mysqlbackup` indicated it was using TLSv1.2. (Bug #30268505)

- Backups for a server using the keyring_file or keyring_encrypted_file plugin failed with the error `Opening of file master_keyring_kef failed` if the keyring file was located in the server's data directory. This was because in that case, the plugin returned a path for the keyring file that was relative to the data directory, with which `mysqlbackup` could not locate the file. With this fix, the plugin returns a full path of the file to `mysqlbackup`. (Bug #30238406)

- `mysqlbackup` quit unexpectedly when the `validate` command was issued without specifying any command options. With this fix, `mysqlbackup` quits gracefully in the situation by throwing a proper error. (Bug #30204114)

- The default value of the `--page-reread-time` option was 0 millisecond, instead of 100 milliseconds as documented in the manual. (Bug #30036877)

- When a `copy-back-and-apply-log` operation was applied on a prepared backup, the warning `Apply-log operation has already been done on that backup` appeared twice in the output of `mysqlbackup`. (Bug #29941423)

- A backup sometimes failed with `mysqlbackup` reporting that an undo log file looked corrupted when the system variable `innodb_undo_log_encrypt` had been set to `ON` on the server. It was because the encryption information had not yet propagated to the undo log file hearer when the file was copied. With this fix, in the situation, `mysqlbackup` waited until an undo log file's header is updated before copying it. (Bug #29545236)

- When restoring an incremental backup, `mysqlbackup` deleted tables on the server that were not included in the incremental backup. (Bug #29399666)

- A backup failed if the server has two external undo tablespaces on different file paths but with the same file name. This was because `mysqlbackup` copied all undo tablespaces into the same directory during a backup, causing a file name conflict. With this fix, when copying undo tablespaces, the pathname of a tablespace relative to `innodb_undo_directory` is preserved, so there will be no more file name clashes. (Bug #29340016)

- 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)

Changes in MySQL Enterprise Backup 8.0.18 (2019-10-14, General Availability)

MySQL Enterprise Backup 8.0.18 is the latest release for MySQL Enterprise Backup. It only supports MySQL Server 8.0.18. For earlier versions of MySQL 8.0, use the MySQL Enterprise Backup version with the same version number as the server. For MySQL server 5.7, please use MySQL Enterprise Backup 4.1, and for MySQL Server 5.6 and 5.5, please use MySQL Enterprise Backup 3.12.

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- Close to the end of a backup, `mysqlbackup` applied the FLUSH TABLES tbl_name [, tbl_name] ... WITH READ LOCK statement on all non-InnoDB tables. With this fix, the statement is not applied to tables that are not included in a partial backup, thus avoiding unncessary locks for some tables. (Bug #29873048)

- The International Components for Unicode (ICU) library for regular expressions, used by MySQL Server 8.0, has now replaced the RE2 library as the library used by MySQL Enterprise Backup for handling regular expressions.

  This change also removes a memory leak associated with the use of the Partial Backup and Restore Options. (Bug #27374240, Bug #29840790)

- The `image-to-backup-dir` command is now an alias for the `extract` command.

- Two new options, `--compression-algorithms` and `--zstd-compression-level`, have been introduced for configuring compression for server connections. See Connection Options and Command Options for Connection Compression for details.

- MySQL Enterprise Backup now supports a faster way to create incremental backups by using the page tracking functionality on MySQL Servers. To use this new feature, set `--incremental=page-track`. See Incremental Backup Using Page Tracking for details.

- The `--uncompress` option is now supported for the `extract` operation: Files from a compressed single-file backup can now be extracted and uncompressed with a single command.

Bugs Fixed

- Backups on a Group Replication primary node failed when `mysqlbackup` tried to upgrade the `mysql.backup_history` table on the node (see Backup History Table Update for details), and there was a problem causing the node to switch to `super_read_only` mode. With this fix, `mysqlbackup` checks if the node is in `super_read_only` mode during the `backup_history` table upgrade and, if so, skips the steps in the upgrade process that caused the problem. (Bug #30065637)

- After a partial backup created with the `--only-innodb` option was restored, the server failed to start due to an assertion error if the clone plugin or thread pool plugin was loaded during the server startup. It was because the backup created with the `--only-innodb` option did not include the performance schema, and the plugin was looking for the data folder for the performance schema when the restored server was being started. This fix prevents the problem by having `mysqlbackup` create an empty `performance_schema` folder under the backup's data directory when creating backups with `--only-innodb`. (Bug #29999075)

- Wrong value for the `--safe-slave-backup-timeout` option was printed when the `--help` option was used with `mysqlbackup`, if `--safe-slave-backup-timeout` was set with an unsigned integer. (Bug #29994968)
- A backup failed for a server containing encrypted InnoDB tables if the server was started with `--skip-grant-tables`. It was because the server only accepted connection through a Unix socket in that case, but `mysqlbackup` did not pass the value of `--keyring-migration-socket` to the keyring migration server; this fix makes `mysqlbackup` pass the option whenever it connects to a server to be backed up with a socket. (Bug #29954367)
- When restoring a full backup of a slave server in a replication setup, `mysqlbackup` gave warnings for missing relay log information file and master information file even if those files were never used on the backed-up server. This fix removes the unnecessary warnings. (Bug #29941160)
- When the `--compress-method` option was used at an `apply-log` operation, the operation failed, and the backup was corrputed. With this fix, `mysqlbackup` throws an error and quits whenever any compression options are used with `apply-log`. (Bug #29941117)
- At the end of any apply-log operation, `mysqlbackup` printed the message `INFO: Backup was originally taken with the --include regexp option` whenever a partial backup option that made use of regular expressions was used during the backup. With this fix, under the situation, `mysqlbackup` simply mentions in the log message that the backup is a partial one. (Bug #29872975)
- When a base backup was restored with `--skip-relaylog` but a subsequent incremental image backup restore did not use the option, the relay log files were copied from the incremental image backup onto the server. With this fix, an incremental image restore, with or without using the `--skip-relaylog` option, does not restore the relay log files whenever its base backup did not restore the relay log. (Bug #29864964)
- A memory leak that occurred with backups involving encrypted redo log has been removed. (Bug #29841265)
- A memory leak caused by the use of the `--optimistic-time` option has been removed. (Bug #29841031)
- When a base backup was restored with `--skip-binlog` but a subsequent incremental image backup restore did not use the option, the binary log files were copied from the incremental image backup onto the server, and the server could not be started because of the incomplete binary log. With this fix, an incremental image restore, with or without using the `--skip-binlog` option, does not restore the binary log files whenever its base backup did not restore the binary log. (Bug #29802632)
- `mysqlbackup` hung when performing an incremental or compressed image backup when the `--limit-memory` option was set to a low value. This fix prevents the problem by adjusting the way data buffers are allocated in the situation. (Bug #29773223)
- `mysqlbackup` created some duplicate entries in the binary log index file if the binary log was stored outside of the data directory on the backed-up server. (Bug #29564487)
- When restoring an incremental backup created with the `--skip-final-rescan` option, the external tablespaces on the target server got deleted. It was due to the missing entries for the external tablespaces in the incremental backup's tablespace_tracker file, and this fix corrects the issue. (Bug #29513642)
- Attempts to extract binary log files from a compressed backup failed with a `No such file or directory` error when the size of the binary log files on the backed up server was greater than 16MB. (Bug #28787312)

Changes in MySQL Enterprise Backup 8.0.17 (2019-07-22, General Availability)

MySQL Enterprise Backup 8.0.17 is the latest release for MySQL Enterprise Backup. It only supports MySQL Server 8.0.17. For earlier versions of MySQL 8.0, use the MySQL Enterprise Backup version with the same version number as the server. For MySQL server 5.7, please use MySQL Enterprise Backup 4.1, and for MySQL Server 5.6 and 5.5, please use MySQL Enterprise Backup 3.12.

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- Before the current release, when backing up a server that used the keyring_okv plugin for InnoDB table encryption, `mysqlbackup` must not be run by a sudo user of its operating system. This restriction has now been removed. (Bug #29020232)
- The `--datadir` option is no longer required for restoring a TTS backup. If the option is specified and its value does not match with that of the target server, the restore will be aborted. (Bug #28546760)
- The `--incremental-base` option now accepts a new value, `history:last_full_backup`, which makes it easy to create a differential backup. See the description of `--incremental-base` for details.
- To avoid `mysqlbackup` failing to catch up with the growing redo log during a backup operation and missing redo log data, `mysqlbackup` now utilizes redo log archiving, a new feature available on MySQL Server 8.0.17. Redo log archiving can be disabled using the new `mysqlbackup` option `--no-redo-log-archive`. See Backing up Using Redo Log Archiving for details.
- `mysqlbackup` now supports encrypted InnoDB redo logs. The encrypted redo tablespaces are handled the same way as the encrypted tablespaces for InnoDB tables. See Working with Encrypted InnoDB Tablespaces for details.

Bugs Fixed

- A backup failed when the value of the server's system variable `innodb_undo_directory` contained in itself the file path for the server's data directory. It was due to a mishandling of the file path prefix of the undo tablespace directory by `mysqlbackup`, which has been corrected by this fix. (Bug #29849566)
- Restore of an incremental backup failed if its base full backup had been restored with the `--skip-binlog` option. (Bug #29757701)
- When the binary log on the server was more than one level below the data directory on the directory tree, `mysqlbackup` failed to copy the binary log into a backup. This was due to an error on parsing the path of the binary log directory, which has been corrected by this fix. (Bug #29710251)
- If a relative path was used with the `backup_innodb_data_home_dir` option when backing up a server, the whole directory specified by the option was being copied into the target server's data directory during a restore of the backup. Not only was that not the expected behavior of `mysqlbackup`, but it also made subsequent backups of the server failed when the same argument for `backup_innodb_data_home_dir` was used again. (Bug #29613025)
- During a backup operation, when a table or database name contains a slash (/), `mysqlbackup` always treated the corresponding tablespace as an external tablespace; if that was not actually the case, restore for the backup was going to fail. With this fix, `mysqlbackup` checks if the tablespace is really external and then handles it appropriately. (Bug #29472939)

- External undo tablespaces were missing after a restore of a backup directory extracted from a backup image using the `image-to-backup-dir` command. It was because of the mishandling of the tracker file for external tablespaces by the `image-to-backup-dir` command, which has been corrected by this fix. (Bug #29401027)

- While no upgrade path exists between MySQL Enterprise Backup 4.1 and 8.0, an attempt to update the `mysqlbackup` package from release 4.1.3 to 8.0.16 on Ubuntu failed with the complaint that the update tried to overwrite the installation directory for `mysqlbackup`. With this fix, package conflict information has been added so that at the same attempt, the old package is uninstalled (with the user's consent) before the new package is installed. (Bug #29314267)

- When using MySQL Enterprise Backup 8.0 to back up MySQL Server 5.7, an error was thrown, and the error message suggested a wrong version of MySQL Enterprise Backup to be used for the Server. With this fix, the appropriate version of MySQL Enterprise Backup is suggested. (Bug #29195233)

- When backing up a server that used the keyring_okv plugin for InnoDB table encryption, if the `--host`, `--user`, and `--port` options were not specified with the mysqlbackup command via the command line or a configuration file, the backup failed. It was because in that case, `mysqlbackup` had no values for those options it could use to connect to the server that took care of keyring operations. With this fix, default values are now set, so that `mysqlbackup` connects to the server on `localhost` as `root` and on port `3306` for keyring operations when those options are not specified. (Bug #29015923)

- A `copy-back-and-apply-log` operation for a compressed backup created using the `--backup_innodb_data_home_dir option` with a relative file path terminated with signal 6. (Bug #28967141)

- `mysqlbackup` hung during a restore operation when the backup contained more than a hundred InnoDB tablespaces. (Bug #28884254, Bug #29674585)

- A restore operation for a compressed backup failed with an `unexpected end of file` error when the backup was created using `--compress-method=zlib` and the `innodb_page_size` was smaller than 16KB. (Bug #28623215)

- A backup created on an EL7 platform containing InnoDB tables encrypted with MySQL Enterprise Transparent Data Encryption (TDE) could not be restored to a server on a Solaris platform. It was because in this case, the source and the target platforms of the backup used different byte ordering formats, causing difficulties in loading the encryption key from the backup. This fix prevents the issue by adding detection and conversion utilities for different system architectures. (Bug #28569367)

- Using the `--uncompress` option for restoring a backup not created with the `--compress` option caused the operation to fail with the error `No such file or directory`. With this fix, the proper error is thrown in the situation. (Bug #28334690)

- A backup failed with the error `Log scan was only able to reach...` when there was a large amount of DML activities occurring in parallel on the server that was being backed up. (Bug #27555969)

- During the InnoDB buffer pool dump in a backup operation, `mysqlbackup` sometimes reported failure for the dump while it was actually still in progress. The fix prevents the problem by improving the way `mysqlbackup` checks for the status of the dump. (Bug #27185901)

Changes in MySQL Enterprise Backup 8.0.16 (2019-04-25, General Availability)

MySQL Enterprise Backup 8.0.16 is the latest release for MySQL Enterprise Backup. It only supports MySQL Server 8.0.16. For earlier versions of MySQL 8.0, use the MySQL Enterprise Backup version with

the same version number as the server. For MySQL server 5.7, please use MySQL Enterprise Backup 4.1, and for MySQL Server 5.6 and 5.5, please use MySQL Enterprise Backup 3.12.

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- `mysqlbackup` now supports encrypted InnoDB undo logs. The encrypted undo tablespaces are handled the same way as the encrypted tablespaces for InnoDB tables. See Working with Encrypted InnoDB Tablespaces for details.
- Near the end of the backup process, instead of locking the whole server instance for a brief period of time, `mysqlbackup` now applies these locks consecutively:

  1. A backup lock on the server instance, which blocks DDLs (except those on user-created temporary tables), but not DMLs on InnoDB tables.
  2. A `FLUSH TABLES tbl_name [, tbl_name] ... WITH READ LOCK` operation on all non-InnoDB tables, for copying the relevant ones among them into the backup. This step is skipped if no user-created non-InnoDB tables exist.
  3. A brief blocking of logging activities on the server, for collecting logging-related information.

  See The Backup ProcessThe Backup Process for details. The removal of the lock on the whole server instance reduces disruption to the database service by the backup operation.

  

  Important

  The change requires that the `BACKUP_ADMIN` and `SELECT` privileges on all tables be granted to the user by which `mysqlbackup` connects to the server (the `BACKUP_ADMIN` privilege is automatically granted to users with the `RELOAD` privilege when an in-place upgrade to MySQL Server 8.0 from an earlier version is performed).

- `mysqlbackup` now supports dynamic changes to undo tablespaces on the server being backed up. During a restore, the default undo tablespaces, as well as any non-default undo tablespaces resided in the backed-up server's data directory, are restored to the location pointed to by the `mysqlbackup` option `--innodb_undo_directory`. Non-default, external undo tablespaces are restored to the locations they were found on the backed-up server. See undo log files for details.
- In addition to the requirement that the target data directory for a restore specified by the `--datadir` option must be non-existent or empty, `mysqlbackup` now enforces the same rule for the `--innodb_data_home_dir`, `--innodb_log_group_home_dir`, and `--innodb_undo_directory` options (the `--force` option cannot be used to override the requirement on the three options).

Bugs Fixed

- Zip packages of `mysqlbackup` contained duplicate files, which have now been removed. (Bug #29497272, Bug #94683)
- `mysqlbackup` might quit unexpectedly if it lost its connection to the server at the middle of a backup operation. With this fix, `mysqlbackup` exits gracefully in the situation after throwing the appropriate errors. (Bug #29376006)

- `mysqlbackup` returned success for a backup to Oracle Cloud Infrastructure Object Storage Classic even after the backup failed. (Bug #29362469)
- Restore of an incremental backup failed if, on the server, some binary log files had been purged in between the times the incremental backup and its base backup were made. (Bug #29306026)
- A backup failed for `mysqlbackup` if the path given by the `--backup-dir` option was of the Universal Naming Convention (UNC) format, as `mysqlbackup` failed to create the backup directory then. (Bug #29190803)
- A `mysqlbackup` operation failed when the `backup-image` option was supplied to `mysqlbackup` in a configuration file instead of on the command line. (Bug #29157495)
- 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)
- `mysqlbackup` quit unexpectedly when the `--password` option was used twice, with no argument given at the second time, in a `mysqlbackup` command that was itself invalid aside from the use of the `--password` option. (Bug #28894102)
- When a compressed incremental folder backup was restored, the binary log files created in between the times of the base and the incremental backups were not copied onto the target server. (Bug #28773998)
- A restore operation for a database containing encrypted InnoDB tables failed without returning a proper error message when the the `--encrypt-password` option was not used in the `mysqlbackup` command. (Bug #28773077)
- After restoring an incremental backup taken from a MySQL Community Server with encrypted InnoDB tables, the keyring file of the restored server became corrupted, so the server could not be started. (Bug #28422191)
- A restore operation could corrupt a backup when, by mistake, a user specified the source directory to become the target directory for restoring some files (for example, specifying what was the backup's `--backup_innodb_data_home_dir` value as the restore's `--innodb_data_home_dir` value). This fix prevents the problem by having `mysqlbackup` throw an error when the command options make the source and target file paths the same for any file copying during a restore. (Bug #28376873)
- 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)
- A restore operation failed with the error that the binary log index file could not be opened if the binary log base name for the backed-up server was a substring of the word “index”. (Bug #28127023)
- `mysqlbackup` tried to connect to a remote host specified by the `--host` option, while it was supposed to ignore the option (see Connection Options for details). With this fix, the option is now ignored.

  

  Note

  As a side-effect of this change, on Unix-like platforms, `mysqlbackup` command that used the `--host` option now needs to use the `--protocol=TCP` option to indicate explicitly that `mysqlbackup` is to connect to `localhost` using TCP/IP.

  (Bug #25911987)

- When the option `--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 the option was used. With this fix, log flushing was always performed to prevent the error. (Bug #25412655)

Changes in MySQL Enterprise Backup 8.0.15 (2019-02-01, General Availability)

MySQL Enterprise Backup 8.0.15 is the latest release for MySQL Enterprise Backup. It only supports MySQL Server 8.0.15. For earlier versions of MySQL 8.0, use the MySQL Enterprise Backup version with the same version number as the server. For MySQL server 5.7, please use MySQL Enterprise Backup 4.1, and for MySQL Server 5.6 and 5.5, please use MySQL Enterprise Backup 3.12.

This release contains no functional changes and is published to align version number with the MySQL Server 8.0.15 release.

Changes in MySQL Enterprise Backup 8.0.14 (2019-01-21, General Availability)

MySQL Enterprise Backup 8.0.14 is the latest release for MySQL Enterprise Backup. It only supports MySQL Server 8.0.14. For earlier versions of MySQL 8.0, use the MySQL Enterprise Backup version with the same version number as the server. For MySQL server 5.7, please use MySQL Enterprise Backup 4.1, and for MySQL Server 5.6 and 5.5, please use MySQL Enterprise Backup 3.12.

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- `mysqlbackup` now supports encrypted binary and relay log. See descriptions for the undo log files for details.
- `mysqlbackup` now supports the `--ssl-fips-mode` option, which controls whether `mysqlbackup` operates in FIPS mode. See FIPS Support for details.

Bugs Fixed

- When working with a Group Replication cluster, `mysqlbackup` quit unexpectedly near the end of a backup operation when, in order to write to the `backup_history` table, it tried to connect with an unencrypted connection to one of the nodes on which the backup user had not logged on before. It was because, as a user created with the caching_sha2_password plugin (enabled by default on MySQL 8.0 servers), the backup user must log on with an encrypted connection when it connects to the server for the first time; the attempt to log on thus failed, and `mysqlbackup` could not handle the failure. With this fix, at such failures, `mysqlbackup` quits gracefully with the warning that the backup operation is finished without updates to the backup history. (Bug #28893180)
- An `apply-incremental-backup` operation failed with an error (`RDR1 ERROR: Unable to remove relaylog files from full backup`) when the incremental backup was created with the `--compress` option. (Bug #28366241)
- `mysqlbackup` quit unexpectedly during an `apply-incremental-backup` operation if the backed up server had been started using relative paths for `--datadir` and `--log-bin`. (Bug #28334521)

- Attempts to restore a backup of a MySQL 5.7 Server to a MySQL 8.0 Server resulted in a strange error message (`Server_version is not obtained`). With this fix, `mysqlbackup` now indicates that the operation is not supported. For related information, see Restoring a Backup with a Database Upgrade or Downgrade. (Bug #27952379)

- After restoring an incremental folder backup and putting its binary log at a specified location different from that for the base backup, the older binary log files of the base backup were not removed by `mysqlbackup`. (Bug #27890472)

- `mysqlbackup` quit unexpectedly when backing up a MySQL Server of release 8.0.12 or later for the first time if the `ALTER` privilege on the `mysql.backup_history_new` table had not been granted to the MySQL user with which `mysqlbackup` connected to the server. With this fix, `mysqlbackup` quits gracefully in the situation after throwing the proper error.

  Also, the `CREATE`, `INSERT`, and `DROP` privileges on `mysql.backup_history_old` and `CREATE`, `INSERT`, `DROP`, and `ALTER` privileges on `mysql.backup_history_new` are now required only for backing up for the first time a MySQL Server that has been upgraded from 8.0.11 or earlier and has been backed up by MySQL Enterprise Backup before. (Bug #27879530, Bug #28546256)

- Partial backups sometimes failed because full-text index files had their file names matched by the regular expression provided by the `--include-tables` option, and the files were then handled as ordinary tablespace files by `mysqlbackup`. With this fix, `mysqlbackup` excludes any full-text index files from backups. (Bug #25044900)

Changes in MySQL Enterprise Backup 8.0.13 (2018-10-22, General Availability)

MySQL Enterprise Backup 8.0.13 is the latest release for MySQL Enterprise Backup. It only supports MySQL Server 8.0.13. For earlier versions of MySQL 8.0, use the MySQL Enterprise Backup version with the same version number as the server. For MySQL server 5.7, please use MySQL Enterprise Backup 4.1, and for MySQL Server 5.6 and 5.5, please use MySQL Enterprise Backup 3.12.

- Functionality Added or Changed

- Bugs Fixed

Functionality Added or Changed

- `mysqlbackup` now supports backup compression (the use of the `--compress` and `--uncompress` options) for incremental backups (except for incremental backups created with the `--incremental-with-redo-log-only` option).

- `mysqlbackup` now supports transparent page compression for InnoDB tables. The support is enabled by setting the `mysqlbackup` option `--compress-method=punch-hole`; see description for the option for details.

Bugs Fixed

- Restoring an incremental backup image using the `copy-back-and-apply-log` command failed with `mysqlbackup` complaining that the server repository configuration (including, for example, value of `innodb_data_file_path`) was unknown for the target server. With this fix, `mysqlbackup` gets the required information from the `backup-my.cnf` file already restored with the base backup of the incremental backup. (Bug #28411028)

  References: This issue is a regression of: Bug #27429244.

- `mysqlbackup` hung when a backup operation failed due to a full disk. With this fix, `mysqlbackup` quits gracefully in the situation by throwing an error. (Bug #28399821)
- During an `--apply-incremental-backup` operation, `mysqlbackup` attempted to delete the binary log of the backed-up, running server. (Bug #28377502)
- On FreeBSD platforms, using the `--show-progress` option did not make `mysqlbackup` print progress reports. (Bug #28350122)
- A `mysqlbackup` operation on an image stored on an OpenStack cloud storage service sometimes failed with a segmentation fault or a bad URL error. It was because of a race condition caused by an uninitiated variable, which has been eliminated by this fix. (Bug #28189239, Bug #28183729)
- Backups for databases with encrypted InnoDB tables failed when the `--compress` option was used. (Bug #28177466)
- The Windows version of MySQL Enterprise Backup did not display its build ID when invoked. (Bug #27916702)
- A `mysqlbackup` operation on an image stored on an OpenStack cloud storage service failed with a `401 Unauthorized` error when the operation took a long time and the authentication token for the cloud access expired. With this fix, a separate thread in `mysqlbackup` requests a new token from the OpenStack cloud service in that situation, so that the operation can continue. (Bug #27893174)
- When the `--show-progress=table` option was used, `mysqlbackup` gave a warning in the error log on an aborted connection to the server near the end of the operation. It was because the connection to the server for writing to the `backup_progress` table had remained open. With this fix, the connection is properly closed after the `mysqlbackup` operation is finished. (Bug #27647283)
- When an incremental backup was restored without using the `--log-bin` option, the binary log was not restored to its original location on the backed up server, but to the location specified by `--log-bin` earlier during the restore of the base backup. The same occurred for relay logs of incremental backups for slaves when the `--relay-log` option was not used. (Bug #27545745)
- If, when a backup was in progress and `mysqlbackup` was reading the binary log (or the relay log) index file and the server tried to modify the index file (because, for example, a log flush or log purge just took place), the binary logging (or relay logging) stopped; the server also quit unexpected on Windows platforms. It was because `mysqlbackup` did not handle well the file sharing violation. With this fix, `mysqlbackup` now reads the index file using the local file system API, which handles the file sharing violation gracefully; also, `mysqlbackup` now copies the index file into its buffer and then closes it, instead of keeping it open for long, so the server can modify or delete the index file more easily. (Bug #22914974, Bug #26047119)

Changes in MySQL Enterprise Backup 8.0.12 (2018-07-27, General Availability)

MySQL Enterprise Backup 8.0.12 is the latest release for MySQL Enterprise Backup. It only supports MySQL Server 8.0.12. For earlier versions of MySQL 8.0, use the MySQL Enterprise Backup version with the same version number as the server. For MySQL server 5.7, please use MySQL Enterprise Backup 4.1, and for MySQL Server 5.6 and 5.5, please use MySQL Enterprise Backup 3.12.

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- **Important Change:** Starting from release 8.0.12, the storage engine of the `mysql.backup_history` table on a backed-up server has switched from CSV to InnoDB, and a new column for server UUIDs has been added to the table. See Backup History Table Update for the new user privileges required by `mysqlbackup` due to this change.
- **Important Change:** When working with a Group Replication setup, `mysqlbackup` now makes the backup history available to all members of the server group by making sure that the `backup_history` table is updated on a primary node after each `mysqlbackup` operation. See Using MySQL Enterprise Backup with Group Replication for details.

  With the implementation of this feature, the new user privilege of `SELECT` on `performance_schema.replication_group_members` is now required by `mysqlbackup` to work with any server, even when it does not belong to a Group Replication setup. See Grant MySQL Privileges to Backup Administrator for details.
- Version information for `mysqlbackup` is now printed to the `stdout` instead of the `stderr` stream when the `--version` or `--help` option is used. (Bug #27253989)
- OAuth is now supported for Oracle Cloud Storage client authentication. Two new options, `--cloud-storage-url` and `--cloud-oauth-token`, have been introduced for the purpose. See Cloud Storage Options for details.

Bugs Fixed

- Backups for a server failed when it had `ANSI_QUOTES` in its values for `sql_mode`. (Bug #27939774)
- The maximum value that could be set for the `--safe-slave-backup-timeout` option was 2700 (seconds), which automatically replaced any larger value. With this fix, there is no longer a maximum limit, even though a high value is not recommended; see the description of `--safe-slave-backup-timeout` for details. (Bug #27883020)
- Restoring an incrementation backup on top of a data directory restored using a compressed backup failed. It was because `mysqlbackup` did not set `is_compressed=0` in the `backup_variables.txt` file inside the restored data directory. (Bug #27787988)
- When `mysqlbackup` performed sanity checks on InnoDB tablespaces and found a space ID mismatch for an FSP header and a page header, the name of the problematic tablespace was not given in the error report. (Bug #27752703)
- If an ALTER TABLE statement was executed on the server before an incremental backup was taken, a server restored with the backup on which the incremental backup was applied (using the `apply-incremental-backup` command) could not be started, as the `.ibd` file of the altered table was missing in the restored data. (Bug #27735134)
- After a server has been restored using an incremental backup created with the `--incremental-with-redo-log-only` option, it could not be started. (Bug #27722525)
- `mysqlbackup` issued a warning whenever the number of files specified in the system variable `innodb_data_file_path` of the server to be backed up exceeded 100. With this fix, a warning is issued only if the number of InnoDB data files to be opened exceeds the number specified by the system variable `innodb_open_files`. (Bug #27701402)
- Backups failed for a server that had been started with a value for `--innodb_log_file_size` different from the one the server was initiated with. (Bug #27571663)

- An `apply-incremental-backup` operation failed when individual tablespaces with relative file paths were involved. (Bug #27278876)
- In a Group Replication setting for MySQL servers, when changes were made to one group member and a backup was taken on another, the relay log for the replication applier was missing from the backup, so that the restored server could not be started. (Bug #25534078)
- When a compressed backup was restored with the `--innodb_data_home_dir` option pointing to a restore location outside of the data directory, `.ibd` files were still being copied into the data directory, causing an exception to be thrown at the attempt to start the restored server. (Bug #24826986)

Changes in MySQL Enterprise Backup 8.0.11 (2018-04-19, General Availability)

MySQL Enterprise Backup 8.0.11 is the latest release for MySQL Enterprise Backup. It only supports MySQL Server 8.0.11. For MySQL server 5.7, please use MySQL Enterprise Backup 4.1, and for MySQL Server 5.6 and 5.5, please use MySQL Enterprise Backup 3.12.

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- Offline backups are no longer supported by `mysqlbackup`. As a result, a number of options used for offline backup operations have been removed. See What's New in MySQL Enterprise Backup 8.0? for details. (Bug #27429244)
- The server option `--secure-auth`, deprecated since MySQL 5.7.5, is no longer supported by `mysqlbackup`. (Bug #27265328)
- Servers' use of the keyring_encrypted_file and keyring_aws plugins is now supported. See Working with Encrypted InnoDB Tablespaces for details. (Bug #27127898)
- 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)
- The relay log for a backed-up slave server, instead of being restored always to the data directory on the target slave server, is now restored by default to the same location it was found on the backed-up slave server. It can also be restored to a different location specified with the new `--relay-log` option. (Bug #25141738, Bug #83927)
- The binary log for a backed-up server, instead of being restored always to the data directory on the target server, is now restored by default to the same location it was found on the backed-up server. It can also be restored to a different location specified with the new `--log-bin` option. (Bug #25141738, Bug #83927)
- The `backup_history` table now includes a `server_uuid` column, which stores the value of the `server_uuid` of the backed up server.
- The options `--ssl` and `--ssl-verify-server-cert`, already deprecated in MySQL Enterprise Backup 4.1, have now been removed. Use the `--ssl-mode` option instead to configure the security mode of your connection to the server.
- MySQL Enterprise Firewall is now supported.
- A new option, `--tls-version`, specifies the protocols `mysqlbackup` permits for encrypted connections to MySQL servers.

- A file now tracks information of external tablespaces for a backup or restore in JSON format. See description for tablespace_tracker in Types of Files in a Backup for details.
- `mysqlbackup` could not restore the auto increment values in tables and the corruptions flags for indexes onto a server. The tasks are now made possible by having `mysqlbackup` copying onto the target server blocks of redo logs that cover the duration from the latest checkpoint up to the backup end time, so that the restored server can, during the recovery phase at its first start, restore the auto increment values and the corruption flags using those blocks.
- The buffer size for cloud transfers can now be specified using the new `--cloud-buffer-size` option. See Cloud Storage Options for details.
- HTTP Basic Authentication and non-chunked transfer are now supported for backup and restore using OpenStack Swift-compatible object storage services. Two new options, `--cloud-basicauth-url` and `--cloud-chunked-transfer`, have been introduced for these purposes. See Cloud Storage Options for details.

Bugs Fixed

- After restoring a full backup, if the following restore of an incremental backup changed the restore location of the undo log, either `mysqlbackup` hung, or the restored server failed to start. With this fix, `mysqlbackup` quits with a proper error (“Undo tablespace in the base backup not found”) in the situation.

  Users should make sure the undo log location does not change between successive restores of a full and an incremental backups, or of two incremental backups. (Bug #27530916)
- When restoring an incremental backup containing encrypted InnoDB tables to a MySQL Community Server, the password provided to `mysqlbackup` with the `--encrypt-password` option was never validated, so that when the wrong password was given, the restore still succeeded, but the restored server could not be started. With this fix, `mysqlbackup` throws an error and stops the restore if the password is wrong. (Bug #27483449)
- `mysqlbackup` failed to backup to an Amazon S3 cloud storage. (Bug #27231229)
- An `apply-incremental-backup` operation corrupted the non-InnoDB files in its target backup when the sizes of those files are smaller in the incremental backup than in the target backup. (Bug #27001934)
- `mysqlbackup` could not establish a connection to the server with a Unix socket using the option `--protocol=SOCKET`. (Bug #26977679)
- A number of memory leaks were observed when running `mysqlbackup`. They have now been fixed. (Bug #26495834, Bug #26373259, Bug #26093563, Bug #26423820, Bug #26497245)
- After applying an incremental backup created with the `--incremental-with-redo-log-only` option to a full backup, the full backup's binary log became corrupted. (Bug #26403452)
- On macOS, `mysqlbackup` failed to determine the relay log file name correctly and thus could not back up the relay log for a slave server. (Bug #25574605)
- `mysqlbackup` only accepts values for `--ssl-mode` in upper case. With this fix, the option's value has become case insensitive. (Bug #25548088)

Index

Symbols

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

R

S

T

U

V

W