MySQL Enterprise Backup 3.9 Release Notes

Abstract

This document lists the changes to the MySQL Enterprise Backup 3.9 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 3.9 documentation, see the MySQL Enterprise Backup User's Guide (Version 3.9.0).

For legal information, see the Legal Notices.

Document generated on: 2015-07-03 (revision: 6424)

Table of Contents

Preface and Legal Notices

This document lists the changes to the MySQL Enterprise Backup 3.9 product, beginning with the most recent release.

Legal Notices

adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software 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 which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. MySQL is a trademark of Oracle Corporation and/or its affiliates, and shall not be used without Oracle's express written authorization. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on 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. 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.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this material is subject to the terms and conditions of your Oracle Software License and Service Agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle or as specifically provided below. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

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.

For more information on the terms of this license, or for details on how the MySQL documentation is built and produced, please visit MySQL Contact & Questions.

For additional licensing information, including licenses for third-party libraries used by MySQL products, see Preface and Legal Notices.

For help with using MySQL, please visit either the MySQL Forums or MySQL Mailing Lists where you can discuss your issues with other MySQL users.

For additional documentation on MySQL products, including translations of the documentation into other languages, and downloadable versions in variety of formats, including HTML and PDF formats, see the MySQL Documentation Library.

Changes in MySQL Enterprise Backup 3.9.0 (2013-08-28)

- Functionality Added or Changed [3]

- Bugs Fixed [3]

Functionality Added or Changed

- Support for the deprecated legacy `ibbackup` and `innobackup` commands has been removed in MySQL Enterprise Backup 3.9.0. Users must use the `mysqlbackup` command to perform MySQL Enterprise Backup operations.

- A new operation, `copy-back-and-apply-log`, allows users to restore a backup from an image file or a backup directory and perform an `apply-log` operation to the restored data to bring them up-to-date, all by a single command. The operation makes the restoration process simpler and faster, and also reduces the disk space required.

- MySQL Enterprise Backup has a new message logging feature, which saves important progress and error information into a log file. The feature is controlled by the new options described in Message Logging Options.

- A new option, `--use-tts`, enables selective backup of InnoDB tables using transportable tablespaces (TTS). This option offers many advantages over the partial backup options available before in earlier versions of MySQL Enterprise Backup.

- A new option, `--skip-unused-pages`, enables the skipping of empty pages during backup operations, reducing the file sizes and thus the required disk space and I/O resources for backups.

- MySQL Enterprise Backup now performs full-instance backups, which, on top of the log files and data files, include all the global variables and plugin details from the backed-up server. A full-instance backup allows the re-creation of the backed-up server's full settings on the server on which the backup is restored. See Files that Are Backed Up for details on the files `server-my.cnf` and `server-all.cnf`, which are newly created to support full-instance backup, and for the function of the file `backup_content.xml` in a full-instance backup.

Bugs Fixed

- The value of the `innodb_undo_directory` option was included in the file `backup-my.cnf` even when the value of `innodb_undo_tablespaces` was set to "0" on the MySQL server being backed up. This fix makes sure that `innodb_undo_directory` is only included in `backup-my.cnf` when `innodb_undo_tablespaces` is not set to "0" on the backed-up server and when `innodb_undo_directory` is set by `mysqlbackup` to something other than the data directory (which is the default value and needs not be included in `backup-my.cnf`). However, due to this change, when restoring a backup made by MySQL Enterprise Backup 3.8.2 or earlier that contains undo logs stored outside of the data directory using MySQL Enterprise Backup 3.9.0 or later, the option `innodb_undo_directory` must be correctly specified if `backup-my.cnf` is used for the `--defaults-file` option. (Bug #17231658)

- If the file `backup_variables.txt` was world writable, an `apply-log` operation did not write to the file the `end_LSN` value at the end of the process. As a result, if an incremental backup was done afterwards, a subsequent `apply-incremental-backup` would fail. (Bug #16951153)

- When using the `--include` option to select InnoDB tables for backup, if there were no matches for the supplied regular expression, the `mysqlbackup` command still finished like normal while no InnoDB tables were copied. This fix makes `mysqlbackup` throw an error and give an error message in such a situation. (Bug #16295721)

- When MySQL Enterprise Backup tried but could not execute a utility specified by the `--exec-when-locked` option, a misleading error message was returned. This fix removes that "mysqlbackup: ERROR: ... " message. (Bug #16267794)

- When performing an `apply-log` operation on the a compressed backup with the `--apply-log` and `--uncompress` options, a warning was returned, saying "... --limit-memory=100 is not sufficient and so ignoring --limit-memory" even when the `--limit-memory` option was not used. This was due to the default memory size of 100 MB allocated for the `apply-log` operation, which is

insufficient for a compressed backup. This fix increases the allocated memory to 300 MB by default, so the warning message will not appear unless the `--limit-memory` option is used to make the memory allocation insufficient. (Bug #16239932)

- Doing a backup with the `mysqlbackup` command could cause an aborted connection warning to appear in the MySQL error log when connection warnings were enabled (with the setting `log_warning=2` on the server). It was because the database connection was not closed correctly after the backup completed. (Bug #15993914, Bug #67876)

- The file `ibbackup_ibd_files` was missing from the `backup_content.xml` file after an incremental backup, and the file `ibbackup_redo_log_only` was also missing from `backup_content.xml` if the backup was performed with the `--incremental-with-redo-log-only` option. (Bug #12554064)

Changes in MySQL Enterprise Backup 3.8.2 (2013-06-18)

Functionality Added or Changed

- MySQL Enterprise Backup has a new progress report feature, which periodically outputs short progress indicators on its operations to user-selected destinations (for example, `stdout`, `stderr`, a file, or other choices). This feature is controlled by the new progress report options described in Progress Report Options.

Changes in MySQL Enterprise Backup 3.8.1 (2013-02-05)

This section documents changes and bug fixes that have been applied in MySQL Enterprise Backup, from version 3.5.1 through version 3.8.1. The 3.8.1 release contains mainly fixes and enhancements compatibility with features of MySQL 5.6.

Functionality Added or Changed

- Normally, the InnoDB system tablespace is extended by one or more megabytes at a time. Under some circumstances, particularly in a disk-full or almost-full situation, MySQL can extend the InnoDB system tablespace by some amount less than a megabyte, and this trailing portion is unused. Now the `mysqlbackup` command backs up such oddly sized system tablespaces even though the size does not precisely match the specified size. This condition causes a warning rather than an error. The unused fractional megabyte is removed from the backup data. (This particular file-size feature applies only to the system tablespace, not to the file-per-table tablespaces in the .ibd files.)

- The `mysqlbackup` command now supports the UNC (Universal or Uniform Naming Convention) syntax for specifying locations on shared disks. This feature lets you start backups to shared drives using Windows Task Scheduler, when shared drives cannot be mapped to a drive letter.

  The UNC syntax for Windows systems has the generic form:

  ```
  \\ComputerName\SharedFolder\Resource
  ```

  You can use either forward or backward slashes in the path names.

  The `mysqlbackup` command does not support the "long UNC" syntax:

  ```
  \\?\UNC\ComputerName\SharedFolder\Resource
  ```

  The UNC path names can be specified with any backup command and option.

  > **Note**
  >
  > Do not use UNC paths to specify the location of InnoDB data files and log files to be backed up. Such files cannot be reliably backed up over a network

> file system. MySQL Enterprise Backup does not issue any warning in this case.

Changes in MySQL Enterprise Backup 3.8.0 (2012-07-27)

This section documents changes and bug fixes that have been applied in MySQL Enterprise Backup, from version 3.5.1 through version 3.8.0. The primary feature of MySQL Enterprise Backup 3.8.0 is performance-related: parallel backup and its associated options `--number-of-buffers`, `--read-threads`, `--write-threads`, and `--process-threads`.

Functionality Added or Changed

- **Performance:** During both backup and restore jobs, certain operations are performed in parallel: reading data, writing data, compressing or decompressing it (if applicable), and validating it (verifying the page checksum during backup). The parallel execution takes advantage of multi-threading. The performance increase is particularly evident when backing up to or restoring from a RAID storage device, and when using compressed backups. See Performance / Scalability / Capacity Options for the options to fine-tune the parallel execution parameters. See Performance Considerations for MySQL Enterprise Backup for general performance advice, including tips for parallel processing for backup and restore.

Changes in MySQL Enterprise Backup 3.7.1 (2012-03-23)

This section documents changes and bug fixes that have been applied in MySQL Enterprise Backup, from version 3.5.1 through version 3.7.1. MySQL Enterprise Backup 3.7.1 is primarily a bug-fix release, with one new feature: the `--incremental-base=history:last_backup` option syntax to simplify taking a sequence of incremental backups.

Functionality Added or Changed

- The `ibbackup` and `innobackup` commands, provided for compatibility with command syntax from earlier releases are deprecated, meaning that they could be removed in a future release. For more information, see Compatibility Information for MySQL Enterprise Backup.

Changes in MySQL Enterprise Backup 3.7.0 (2012-01-04)

This section documents changes and bug fixes that have been applied in MySQL Enterprise Backup, from version 3.5.1 through version 3.7.0.

Functionality Added or Changed

- The single-file backup feature now includes checksum verification to ensure the backup data remains unchanged during any transfers to other systems. Each file within the backup image is tested against a checksum calculated using the CRC32 algorithm, either when files are extracted from the backup image, or using the new `mysqlbackup` option `validate` to test a backup image without extracting. For example:

```
mysqlbackup other_options --backup-image=image_file validate
```

- The `mysqlbackup` option `--start-lsn` is now optional for incremental backups. If you specify the location of an existing full or incremental backup with the new `--incremental-base` option, you can omit the `--start-lsn` option and the `mysqlbackup` command automatically determines the appropriate LSN.

  For safety, the backup job halts with an error if the LSN information cannot be retrieved, or if there is a mismatch between the LSN recorded in the previous backup and the LSN reported by the MySQL instance.

  The automatic LSN detection works with both the original (`--incremental`) and new (`--incremental-with-redo-log-only`) incremental backup techniques.

- The `mysqlbackup` now offers an alternative method of performing incremental backup for InnoDB tables. This technique involves copying the redo log. This backup method is an alternative to the incremental backup of InnoDB tables.

  To enable the redo-log-only incremental backup method, specify the `--incremental-with-redo-log-only` option on the `mysqlbackup` command line, instead of the `--incremental` option).

  > **Note**
  >
  > The benefit of this option is a reduction in random I/O during the backup stage, in favor of sequential I/O. The older incremental backup technique reads only changed blocks from the InnoDB data files. If a high proportion of blocks changed, I/O overhead to read the changed data could limit the performance benefits of an incremental backup. The new technique minimizes this I/O overhead by doing sequential reads to copy the relatively small `ib_logfile*`files. The new technique tends to perform better on systems with big InnoDB tables that receive frequent updates, broadly distributed within the tables. If you have small tables, or relatively few changes, or changes over and over again to the same pages on disk, the old technique might perform better for you. The backup information from the redo logs might also take longer to apply during the apply-log phase. Run benchmarks to determine the best technique for your workload.
  >
  > This technique relies on you having sufficient redo log data to cover the period since the last incremental backup. Because the InnoDB redo log uses a circular buffer where older entries eventually are overwritten, make incremental backups frequently to avoid having older changes unavailable for the incremental backup. Ensure that the value of the `innodb_log_file_size` configuration option is large enough to hold all the changes to InnoDB data that accumulate between incremental backups.
  >
  > When you apply the log data from a series incremental backups to a full backup, the incremental backup data can be produced using either of the techniques. Some incremental backups could be produced by the `--incremental` option and others with the `--incremental-with-redo-log-only` option.
  >
  > Because the incremental backup data format produced by the `--incremental-with-redo-log-only` option is different from the format of MySQL Enterprise Backup 3.6 or 3.5, older `mysqlbackup` versions cannot perform the apply step on a redo-log-only backup. If you use the new incremental backup technique, upgrade the `mysqlbackup` command to the latest MySQL Enterprise Backup level on any machines where you run the apply-log step.
  >
  > This feature is not available in the `ibbackup` command, which is intended only for compatibility with the option syntax from MySQL Enterprise Backup 3.5 and the earlier InnoDB Hot Backup product.

- Performance work within the `mysqlbackup` command makes backup jobs faster with less overhead.

  Performance of backup-related I/O operations is improved, particularly on Windows, by reusing I/O library code and best practices from the MySQL Server product.

  CPU overhead is lessened by reducing the number of memory allocation and deallocation operations.

- A new option for the `mysqlbackup`, `--only-innodb-with-frm`, lets you back up InnoDB tables and their associated `.frm` files with minimal disruption to database processing.

By default, all InnoDB and non-InnoDB tables are backed up, along with all the `.frm` files. While the `.frm` files are being copied, the instance is put into a read-only state. With the `--only-innodb` option introduced in MySQL Enterprise Backup 3.6, only InnoDB tables are backed up, but you must copy the `.frm` files manually, and again this stage happens with the instance in a read-only state. The `--only-innodb-with-frm` is intended for backups where you can ensure that no `ALTER TABLE`, `CREATE TABLE`, `DROP TABLE`, or other DDL statements modify the `.frm` files for InnoDB tables during the backup operation. If the `mysqlbackup` detects that any of the relevant `.frm` files was modified or deleted during the backup job, the command halts with an error.

- When managing backup data using the SBT interface of products such as Oracle Secure Backup, you can change certain settings in the media management software by setting environment variables that are recognized by the SBT library. The new `--sbt-environment` option of the `mysqlbackup` command lets you set such environment variables for the duration of the backup job only, rather than using a wrapper script to set and unset the variables.

  Although the intended purpose of the `--sbt-environment` option is to pass environment variables used by the SBT library, you can set any Unix, Linux, or Windows environment setting this way for the duration of the backup job.

Changes in MySQL Enterprise Backup 3.6.1 (2011-09-28)

Functionality Added or Changed

- MySQL Enterprise Backup can now authenticate to the server being backed up using the Enterprise authentication plugins available in the commercial distributions for MySQL 5.5.16 and higher. For example:

  - With the Windows Native authentication plugin, you can set up a MySQL user ID named the same as the Windows user ID, grant MySQL privileges as described in Grant MySQL Privileges to Backup Administrator, and then perform backups from that Windows account by specifying the `--user` option without a `--password` option.

  - With the PAM authentication plugin, you can connect to the MySQL server using a flexible system to map user IDs and associated privileges.

  For more details about the MySQL pluggable authentication feature, see Pluggable Authentication.

Changes in MySQL Enterprise Backup 3.6.0 (2011-07-01)

This section documents changes and bug fixes that have been applied in MySQL Enterprise Backup, version 3.6. This release has substantial enhancements to `mysqlbackup` syntax and processing over MySQL Enterprise Backup 3.5 and the older InnoDB Hot Backup product. For details, see Compatibility Information for MySQL Enterprise Backup.

Functionality Added or Changed

- The `mysqlbackup` command gains enhanced capabilities to do cold backups, with the `--connect-if-online` option.

- The `mysqlbackup` command can now interface with Media Management Software (MMS) products such as Oracle Secure Backup, using the System Backup to Tape (SBT) protocol.

- The backup operation now is much more "online" than in the past.

  Several new options specify connection information and credentials for the database being backed up.

  The connection-related options are made consistent with the corresponding options used for other MySQL client programs.

You no longer need to construct a dummy configuration file for use with MySQL Enterprise Backup. The `mysqlbackup` command reads options from the standard MySQL configuration file, either from its own `[mysqlbackup]` group or the generic `[client]` group. Details about the layout and locations of files in the MySQL server are retrieved automatically using the database connection, so that you do not need to specify them in the configuration file.

- For simplicity in managing and transferring backup data, you can produce a single-file backup as an alternative to a directory tree of backup files. The single-file backup is a foundational feature that is the basis for other important MySQL Enterprise Backup capabilities, such as streaming the backup data to another server and managing the backup data through a Media Management Software product such as Oracle Secure Backup.

- A new `meta` subdirectory inside the backup data contains information about the backup itself. This metadata is known collectively as the manifest. You can use this information to build additional reporting or management features on top of MySQL Enterprise Backup.

- You can associate comments with each set of backup data, either a single string specified on the command line, or through a separate text file.

- For the fastest backup with the least disruption to MySQL server processing, options such as `--innodb-only` and `--no-locking` let you back up InnoDB tables exclusively. By skipping the backup of non-InnoDB files such as MyISAM tables and `.frm` files, you can avoid the final phase of the backup that waits for other operations in the server to complete, then puts the server into a read-only state.

Changes in MySQL Enterprise Backup 3.5.4 (2011-04-21)

This section documents changes and bug fixes that have been applied in MySQL Enterprise Backup, version 3.5.4.

Bugs Fixed

- Minor fixes for copyright notices.

Changes in MySQL Enterprise Backup 3.5.2 (2010-12-16)

This section documents changes and bug fixes that have been applied in MySQL Enterprise Backup, version 3.5.2.

Functionality Added or Changed

- A call to `posix_fadvise()` can be used to reduce the flush cycle of the operating system cache and improve backup performance. This option is set on by default.

- The combined InnoDB and MyISAM backup functionality of the `innobackup` command is now available on Windows systems. The former Perl script is rewritten in C/C++ as the `mysqlbackup` command. This release continues to include the `innobackup` command, which may be deprecated by the next release. There are also some changes to the syntax as specified in the manual.

- Backup history and progress information is logged to the `mysql.backup_history` and `mysql.backup_progress` tables, so that it can be used by the MySQL Enterprise Monitor product and other tools to easily monitor backup operations. For the details of the backup history table, see Troubleshooting for MySQL Enterprise Backup.

Changes in MySQL Enterprise Backup 3.5.1 (2010-11-01)

This section documents changes and bug fixes that have been applied in MySQL Enterprise Backup, version 3.5.1.

Functionality Added or Changed

- Incremental backup.

- Support for the Barracuda file format of InnoDB. MySQL Enterprise Backup can now backup tables that use recent InnoDB features such as table compression and the `dynamic` row format.