

---

# MySQL Shell Release Notes

## Abstract

This document contains release notes for the changes in each release of MySQL Shell.

For additional MySQL Shell documentation, see <http://dev.mysql.com/>.

Updates to these notes occur as new product features are added, so that everybody can follow the development process. If a recent version is listed here that you cannot find on the download page (<http://dev.mysql.com/downloads/>), the version has not yet been released.

The documentation included in source and binary distributions may not be fully up to date with respect to release note entries because integration of the documentation occurs at release build time. For the most up-to-date release notes, please refer to the online documentation instead.

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

Document generated on: 2017-05-15 (revision: 11665)

## Table of Contents

Preface and Legal Notices .....	1
Changes in MySQL Shell .....	2
Changes in MySQL Shell 1.0.9 (2017-04-12, General Availability) .....	2
Changes in MySQL Shell 1.0.8 (2017-03-06, Release Candidate) .....	3
Changes in MySQL Shell 1.0.5 (2016-09-06) .....	6
Changes in MySQL Shell 1.0.4 (2016-06-23, Milestone 2) .....	6

## Preface and Legal Notices

This document contains release notes for the changes in each release of MySQL Shell.

### Legal Notices

Copyright © 1997, 2017, Oracle and/or its affiliates. All rights reserved.

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

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

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

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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

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

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

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

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

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

## Changes in MySQL Shell

### Changes in MySQL Shell 1.0.9 (2017-04-12, General Availability)

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

#### **Functionality Added or Changed**

- In SQL mode, support was added for `\G` to display the SQL result vertically.

The `--vertical` (or `-E`) command line option was added to display all result sets vertically. This sets the `shell.options.outputFormat` variable to "vertical". (Bug #24848230)

### Bugs Fixed

- The `MYSQLPROVISION` environment variable is no longer supported or required by MySQL Shell. (Bug #25733261)
- `mysqlprovision` copyright text was missing the current year. (Bug #25725965)
- Access to schema objects using property syntax (`db.collection`) was not automatically available. (Bug #25721628)
- The `mysqlprovision` binary was not built by default when compiling from source. (Bug #25699824)
- Executing print with the `--execute` option could cause MySQL Shell to stop unexpectedly. (Bug #25684798)
- SQL import operations would fail if multiple-line comments were present in the SQL being imported. (Bug #25637606)
- When using `dba.configureLocalInstance()`, error messages generated due to a configuration file update failure would not report the cause of failure but a misleading error message of its core code. (Bug #25593140)
- On Windows, error handling was improved to better notify the user when the system's Python does not support SSL. (Bug #25534469)
- `dba.checkInstanceConfiguration()` was not documented in the `dba` global variable help. (Bug #25487579)
- A dissolved cluster object was not being correctly set as invalid, meaning it was possible to access its methods and properties. The fix ensures that dissolved cluster objects are correctly detected and a new error similar to `Can't call function 'function_name'` is generated when executed on a dissolved cluster. (Bug #25141048)
- Global DB object properties could change their semantics at runtime. For example, the property giving the schema name could later be displayed as a table object. This was related to how cached objects were given precedence. Now, on object retrieval, the cache is only accessed if the requested member is not a fixed member of the class. (Bug #24964342)
- The `addInstance()` method did not function on remote MySQL instances. (Bug #24832550)
- MySQL Shell was not detecting changes made manually to the cluster topology to use Group Replication multi-primary mode (`group_replication_single_primary_mode=OFF`). MySQL Shell was not showing the read-write abilities of the instances. (Bug #24756046)
- On Unix, MySQL Shell was showing the password used to run the command in the operating system process list.

## Changes in MySQL Shell 1.0.8 (2017-03-06, Release Candidate)

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

### Functionality Added or Changed

- The URI is now a positional argument when starting MySQL Shell at the command line. (Bug #25077429)
- In the MySQL Shell Python X DevAPI implementation the `mysqlx` and `mysql` modules have been moved into the new `mysqlsh` module. This changes the way you import the modules, now you should issue:

```
mysql-py> from mysqlsh import mysql
mysql-py> from mysqlsh import mysqlx
```

(Bug #25030138)

- XSessions have been removed, which means the `--x` command option is removed. The default session is now a NodeSession. The `mysqlx.getSession()` function is also removed and the `\connect` MySQL Shell command does not accept the `-x` option. (Bug #24958348, Bug #83553)
- The stored sessions functionality has been removed. (Bug #24949016, Bug #83530)
- MySQL Shell version 1.0.8 includes the newly released AdminAPI available in JavaScript and Python which enables you to set up and manage InnoDB clusters. It provides a modern fluent API which wraps the complexity associated with configuring, provisioning and managing an InnoDB cluster, without sacrificing power, flexibility or security.

### Bugs Fixed

- **Linux:** Attempting to create a connection with the wrong user or password on Linux resulted in an `Error Unknown option trace_protocol` message being displayed. The fix ensures that the correct `ERROR: 1045 (28000): Access denied for user` error is displayed. (Bug #25071433, Bug #83759)
- When establishing an SSL connection in MySQL Shell, the SSL mode was passed as a string instead of an integer value. (Bug #25425598)
- MySQL Shell support for connecting to MySQL with SSL was incomplete or did not function correctly in some cases. MySQL Shell now applies SSL parameters correctly. For supported SSL parameters, see [mysqlsh — The MySQL Shell](#). (Bug #25411425, Bug #84546)
- SSL options used to connect to instances were not passed to `mysqlprovision`, which also requires the SSL options to connect to instances. (Bug #25366109)
- `rejoinInstance()` did not use `mysqlprovision`, causing inconsistencies for operations that join the instance to the replicaset. (Bug #25269142, Bug #84240)
- MySQL Shell could exit during `addInstance()` given certain `memberSslCa` values. (Bug #25229928, Bug #84146)
- Executing long multiline SQL queries interactively in SQL mode raised an error. (Bug #25229928)
- Repeated execution of `addInstance()` resulted in the use of SSL for group replication despite SSL being disabled by `memberSsl=false`. (Bug #25226295, Bug #84131)
- When using Python mode, assigning a key element that was identified as a type could cause an unexpected halt. The fix ensures that keys which are strings are correctly identified as such as are interpreted as keys. (Bug #25191539)
- Statements that were not executed, for example due to a syntax error, were not being added to the command history. This has been improved so that any statement is added to the command history and

can be accessed using the up and down cursor keys. Additionally the internal MySQL Shell are now added to the command history. (Bug #24967864)

References: See also: Bug #24669771.

- When MySQL Shell had an open connection and was in SQL mode, pressing **Control-C** caused an unexpected halt. (Bug #24812731)

References: See also: Bug #24663772, Bug #23065126.

- The output of the `\help` command has been updated to include all of the available global objects. (Bug #24764191)
- When using the `shell.prompt()` function, providing an invalid type was not generating the appropriate error. (Bug #24763825)
- The way MySQL Shell processes commands has been improved to properly handle spaces in the command line, and support was added for quoted parameters when spaces are used. To pass a parameter with spaces surround it with double quotes, and if double quotes are part of the parameter they can be escaped using `\`. (Bug #24762017)
- A `cluster.rejoinInstance` operation that added an old node to a new cluster caused MySQL Shell to exit. (Bug #24760911)
- Primitive numeric types and booleans were not being automatically converted between each other following the Python type conversion rules. The fix ensures that a statement such as `dba.verbose=1` is valid, in other words when `a == b` is true, then `b` is assignable to `a` as is typical in Python. (Bug #24744707)
- Using `--log-level=5` or greater wrote passwords to the log as plain text. (Bug #24735878)
- The `--log-level` command option can now use named log levels such as `debug3`. (Bug #24735532)
- In JavaScript mode, switching the metadata connection between repeated calls to `cluster.dissolve()` returned an access denied error message instead of reporting that the cluster is dissolved. (Bug #24674064)
- Support for connecting to a server through MySQL Shell using an account with an expired password was added for Classic and Node sessions. Now, when connecting using an expired password and attempting to perform an operation, a message is returned stating that you must reset your password using `ALTER USER` before executing the statement. (Bug #24673742)
- When an SQL statement followed the terminating character of an SQL comment that ended with `*/` it could result in unexpected behavior because the first character of the statement following the comment was being ignored. The fix ensures inline comments in SQL statements are correctly parsed. (Bug #24671436)
- `cluster.status()` and `cluster.describe()` could give incomplete or incorrect information about the cluster depending on the metadata session connection. (Bug #24621357)
- Results output when using the `--json` option now go inside of a JSON object. (Bug #24575611)
- When passing a node to `cluster.rejoinInstance()`, a number of password prompts were erroneously printed in succession without requiring entry of a password. (Bug #24573793)
- When using the help, space characters at the end of a line were causing the requested help topic to not be found. (Bug #24522689)

- When using SQL mode the usage of single quotes in escaping has been made consistent with the handling in MySQL, for example `SELECT ''' as a;` (Bug #24380669, Bug #82381)
- The help for `xSession` was suggesting SQL mode was available which was incorrect. (Bug #23065163)
- When using SQL mode the command history was removing the semicolon from the end of previously issued lines (Bug #21390737)
- A cluster could be rebooted from a removed instance. (Bug #84766, Bug #25480022)
- On Windows, a backslash character (“\”) could not be used in the the `memberSslCa` value. (Bug #84148, Bug #25230238)
- An invalid read occurred when shutting down a MySQL Shell instance that was started in Python mode. (Bug #83867, Bug #25111564)

## Changes in MySQL Shell 1.0.5 (2016-09-06)

### Bugs Fixed

- Paths to `mysql.js` and `mysqlx.js` were hardcoded and compiled into the MySQL Shell binary, preventing binary packages from being unpackaged to non-standard locations. The `mysql` and `mysqlx` modules for JavaScript are now part of the MySQL Shell application instead of separate files. (Bug #23075933)
- If the connection to a MySQL Server was interrupted, issuing `\status` failed unless followed by a `;`. (Bug #81278, Bug #23213616)
- Columns specified as `TIME` were being displayed by MySQL Shell as `BOOLEAN`. (Bug #81260, Bug #23210710)
- Attempting to access a `Session` object that no longer existed could cause MySQL Shell to halt unexpectedly. In such a situation now an error message is displayed.
- Previously JavaScript modules were imported as follows:

```
var mysqlx=require('mysqlx').mysqlx;
var mysql=require('mysql').mysql;
```

As part of improvements to the way modules are added to MySQL Shell, modules are now imported as follows:

```
var mysqlx=require('mysqlx');
var mysql=require('mysql');
```

## Changes in MySQL Shell 1.0.4 (2016-06-23, Milestone 2)

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

### Functionality Added or Changed

- **Linux:** On Linux, you can now create scripts that are executed by MySQL Shell. See [Executable Scripts](#).
- The `tables` property has been replaced by `getTables()`, which should now be used to access the table list. It is still possible to use `schema.table_name`, but it is only available if the table name is

different from any other member of the Schema object. For example, if a table is called `name` it is not accessible as `schema.name` (which would return the actual schema called `name`) and is only accessible through `schema.getTable('name')`. (Bug #22151376, Bug #22151267)

- The `--session-type` option was removed. To chose the session type using command-line options, use either `--x`, `--node`, or `--classic` instead.
- The `println()` function has been added to the the JavaScript implementation.
- The `\connect_node` and `\connect_classic` commands have been replaced by options that can be used with the `\connect` command. Use `\connect -n` to create a Node session and `\connect -c` to create a Classic session.
- The `mysqlx.getSession()` and `mysqlx.getNodeSession()` functions now support named parameters. For example:

```
mysql-py> session=mysqlx.getSession(host="localhost", dbUser="root", dbPassword="mypass")
mysql-py> session
<XSession:root@localhost:33060>
```

- The `getLastInsertId()` function was renamed to `getAutoIncrementValue()`. The `getLastDocumentIds()` function was added.
- The `isOpen()` function has been added to the Session objects to enable you to check if a connection has been established. A new function `parseUri(String param)` has been added to the `shell` module. It takes a connection string in URI format and returns a `Dictionary` object with the URI components found in the input string.

### Bugs Fixed

- **Microsoft Windows:** The Windows MySQL Shell executable and .dll library were missing version information. This is now shown when you right click on the files and open the Details tab.
- The formatting of the list of connections displayed after issuing `\LSCONN` has been improved to ensure each stored connection is shown on its own line. (Bug #23578574)
- A space was missing after the Python prompt. (Bug #23304556)
- Attempting to create an SSL connection failed with an error and connections were not created. (Bug #23184652)
- Instructions for building MySQL Shell from source are now provided in an `INSTALL` file that is included in MySQL Shell source file download packages. (Bug #23146337)
- Functions depending on an internal cache, such as `Schema.collections`, `Schema.tables`, and `Schema.views` were not being updated correctly, which made statements such as `DROP` appear to not function correctly. (Bug #23095017, Bug #81040)
- `session.schemaname.getTables()` failed if `schemaname` was `uri`. (Bug #22155915)
- `schema.getCollection()` and `session.getSchema()` returned inconsistent results when non-existent names were passed to them. The same error is now returned for both functions. (Bug #21983760)
- Using SSL to create a connection was failing with `MySQL Error (2026): SSL bad version`. (Bug #81094, Bug #23112964)
- The help text has been improved to be more accurate and use correct language. (Bug #81085, Bug #23106440)

- When connecting with a password-less account, an incorrect password warning was being displayed.
- The parsing of JSON input, whether at the command line or piped in, was not detecting incorrect syntax. This could result in error messages being incorrectly identified, particularly when using Classic mode.
- When using the `--execute` command-line option to execute an SQL statement, the statement had to finish with a `;` character.
- If an error occurs when using interactive mode, the traceback is printed. If the traceback contained a `1` character then the traceback was being displayed twice.
- It was not possible to add documents with attributes which started with an uppercase character.
- When using a Classic session and querying a collection, some columns were returned as undefined. The same query using a Node session returned the columns correctly.
- An operation that adds an empty list of documents to a collection, for example `coll.add([ ]).execute()` now always succeeds without an error.