
MySQL Shell 1.0 Release Notes

Abstract

This document contains release notes for the changes in each release of MySQL Shell 1.0.

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 (<https://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 the [MySQL Forums](#), where you can discuss your issues with other MySQL users.

Document generated on: 2019-08-01 (revision: 18502)

Table of Contents

Preface and Legal Notices	1
Changes in MySQL Shell 1.0.11 (2017-11-30, General Availability)	2
Changes in MySQL Shell 1.0.10 (2017-07-28, General Availability)	4
Changes in MySQL Shell 1.0.9 (2017-04-12, General Availability)	9
Changes in MySQL Shell 1.0.8 (2017-03-06, Release Candidate)	10
Changes in MySQL Shell 1.0.5 (2016-09-06)	12
Changes in MySQL Shell 1.0.4 (2016-06-23, Milestone 2)	13

Preface and Legal Notices

This document contains release notes for the changes in each release of MySQL Shell 1.0.

Legal Notices

Copyright © 1997, 2019, 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.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Changes in MySQL Shell 1.0.11 (2017-11-30, General Availability)

- [InnoDB Cluster Added or Changed Functionality](#)

- [InnoDB Cluster Bugs Fixed](#)
- [Bugs Fixed](#)

InnoDB Cluster Added or Changed Functionality

- When you create clusters or add instances you can now override the default group name, local addresses, and group seeds. This makes it easier to customize your clusters. The following options were added to the `dba.createCluster()` and `cluster.addInstance()` commands:
 - use `groupName` with `dba.createCluster()` to set the name of the cluster
 - use `localAddress` to set the address which an instance provides to communicate with other instances
 - use `groupSeeds` to set the instances used as seeds when instances join the cluster

For more information, see [Customizing InnoDB clusters](#). (Bug #26485254, Bug #26838005)

- With the addition of [WL#10611](#) and [WL#10960](#), it was not possible to add or rejoin instances that belonged to a cluster (or a replication group) because `super_read_only=ON` was being set by Group Replication when stopping. To ensure that AdminAPI supports instances running MySQL 8.0.2 and later, the following functions have been modified:
 - `dba.configureLocalInstance()`
 - `dba.createCluster()`
 - `dba.rebootClusterFromCompleteOutage()`
 - `dba.dropMetadataSchema()`

Now, if any of these functions is issued against an instance which has `super_read_only=ON`, in interactive mode you are given the choice to set `super_read_only=OFF`. To force the function to set `super_read_only=OFF` in a script, pass the `clearReadOnly` option set to `true`. For example `dba.configureLocalInstance({clearReadOnly: true})`. For more information see [Super Read-only and Instances](#). (Bug #26422638)

InnoDB Cluster Bugs Fixed

- Account validation did not work correctly unless the session account existed. Now, validation is done using the account that was authenticated by the server. (Bug #26979375)
- The AdminAPI in MySQL Shell for working with InnoDB cluster only supports TCP connections to server instances. The AdminAPI now checks that a TCP connection is in use before starting an operation that requires database access, instead of attempting the operation with another connection type and not succeeding. (Bug #26970629)
- Sometimes the `dba.addInstance()` command failed with an error indicating that the server was in `RECOVERING` state despite being `ONLINE`. The fix ensures the correct instance state is returned. (Bug #26834542)
- If the user running MySQL Shell did not have write permissions to the option file configured by AdminAPI, no error was displayed. (Bug #26830224)
- Using AdminAPI commands on Windows that required SSL resulted in an error due to the Python version being used. (Bug #26636911)
- Creating an InnoDB cluster from an existing Group Replication deployment, by using the `adoptFromGR` option with the `dba.createCluster()` command, would fail with an error stating that the instance was already part of a replication group. The issue was only present in the MySQL

Shell default wizard mode. The fix ensures that the interactive layer of the `dba.createCluster()` command allows the use of the `adoptFromGR` option. (Bug #26485316)

- The warnings generated when creating and adding sandbox instances have been improved. (Bug #26393614)
- When working with instances that had `require_secure_transport=ON`, AdminAPI commands that required a connection to the instance failed. (Bug #26248116)
- The `Cluster.dissolve()` command was trying to stop Group Replication on all of the instances registered in the metadata which lead to connection errors if any of those instances were not contactable, in other words with the state `(MISSING)`. The fix ensures that only instances which can be contacted, in other words with the state `ONLINE`, are stopped. (Bug #26001653)
- When adding instances to an InnoDB Cluster using the appropriate AdminAPI operations, checks are performed to verify the compatibility of any existing tables. If incompatible tables (for example using `MyISAM`) are detected then an error is issued. However the error message was referring to an option not available for the AdminAPI operations: `--allow-non-compatible-tables`. (Bug #25966731)
- The `cluster.rejoinInstance()` command attempted to rejoin an instance even if was already part of the cluster. Now, only instances in the `MISSING` state are accepted by `cluster.rejoinInstance()`. Attempting to rejoin an instance in any other state fails with an error. (Bug #87873, Bug #26870329)
- On Unix, if Python 3 was installed AdminAPI commands failed. (Bug #87731, Bug #26785584)
- When using the `dba.checkInstanceConfiguration()` and `dba.configureLocalInstance()` commands, the account being used was not being checked if it had enough privileges to actually execute the command. The fix ensures that account has the required privileges before proceeding. This also required a change of the privileges given to `clusterAdmin` users. (Bug #87300, Bug #26609909)

Bugs Fixed

- MySQL Shell created a logger but did not deallocate it on exiting the shell. (Bug #25238576)
- Arrays and Objects now accept the `IN` operator. For example:

```
collection.find(" 'Fred' IN username ")
```

Changes in MySQL Shell 1.0.10 (2017-07-28, General Availability)

- [InnoDB Cluster Bugs Fixed](#)
- [Security Notes](#)
- [Functionality Added or Changed](#)
- [Bugs Fixed](#)

InnoDB Cluster Bugs Fixed

- When making cluster topology or membership changes, AdminAPI was not taking into consideration the value of `group_replication_group_name`, which could lead to incorrect, non-deterministic results in scenarios such as a split brain. Now, the following commands validate the InnoDB cluster Metadata and the corresponding instance's `group_replication_group_name` value:
 - `dba.getCluster()`
 - `Cluster.rejoinInstance()`

- `Cluster.forceQuorumUsingPartitionOf()`

If the values of `group_replication_group_name` do not match, the commands abort with an error.

`dba.rebootClusterFromCompleteOutage()` was also updated to ensure that the `group_replication_group_name` variable has not been changed before rejoining the instance. (Bug #26159339)

- AdminAPI now always uses the active user value for the current `mysqlsh` session, whether the value was explicitly specified by the user or is the result of an implicit default used by `mysqlsh`. (Bug #26132527)
- The `addInstance()` function was failing when run against a server using MySQL 8. (Bug #26085799)
- Executing AdminAPI commands on a server with a version of Python lower than 2.7 was failing without the correct error message. (Bug #25975317)
- The checks performed by the AdminAPI upon issuing `dba.rebootClusterFromCompleteOutage()` were more strict than those required by Group Replication. Now, the AdminAPI considers tables with a Primary Key Equivalent (such as a Non Null Unique Key) as compatible, matching the current requirement for Group Replication. (Bug #25974689)
- Executing `dba.createCluster()` on an instance with `sql_mode=ANSI_QUOTES` failed with an error. (Bug #25966057)

References: See also: Bug #25990172.

- When using MySQL Shell on Windows any files created or opened, for example those used during `dba.createSandboxInstance()`, could not be deleted. (Bug #25789094)
- The previous limitation of `auto_increment_increment` being set according to the random 32-bit unsigned integers used for the `server-id` system variable has been resolved. Now, when operating in single-primary mode `auto_increment_increment` is set to 1 and `auto_increment_offset` is set to 2. In multi-primary mode `auto_increment_increment` is set to 7 and `auto_increment_offset` is set to 1 plus the `server_id` modulo 7. (Bug #25724564)
- The randomly generated passwords used by internal users were not compatible with instances running the Password Validation plugin. (Bug #25714751)
- The help for `dba.configureLocalInstance(instance[, options])` has been improved to describe the returned JSON object. (Bug #25703028)
- Issuing `configureLocalInstance()` when using a URI that contained a user without the correct privileges resulted in an incorrect new user being created. Now, if the user in `configureLocalInstance()` URI does not have enough privileges to grant all the necessary privileges for the new user chosen during the interactive wizard configuration the user is not created. (Bug #25614855)
- Issuing `Cluster.rescan()` resulted in non-deterministic behavior which could produce incorrect JSON output, showing an instance that was already part of the cluster as belonging to the `newlyDiscoveredInstances[]` list and to the `unavailableInstances[]` list. This also resulted in MySQL Shell prompting to add or remove the instance from the cluster. (Bug #25534693)
- AdminAPI functions now accept the standard connection parameters as used by `shell.connect`. New validations have been added for when `require_secure_transport` is ON, now it is not possible to create a cluster with `memberSslMode:DISABLED` or to add an instance with `require_secure_transport=ON` to a cluster where `memberSslMode:DISABLED`. (Bug #25532298)

- The parsing of account names, for example when passing the `clusterAdmin` option to `dba.configureLocalInstance()` has been improved. (Bug #25528695)
- Issuing `configureLocalInstance()` twice could fail. (Bug #25519190)
- When passing the `rejoinInstances[]` option to `dba.rebootClusterFromCompleteOutage()`, if no `rejoinInstances[]` option was specified then members were being incorrectly handled during the rebuild. Now, instances that are eligible to be added to the `rejoinInstances[]` list but that are specified in the `removeInstances[]` list are skipped by the interactive wizard that tries to automatically build a `rejoinInstances[]` list if one was not provided. This fix also ensures that both interactive and noninteractive use of MySQL Shell correctly verify the `rejoinInstances[]` list does not contain a unreachable instance. (Bug #25516390)
- When using `dba.deploySandboxInstance()` and passing in `sandboxDir`, the specified path must not exceed 89 characters. (Bug #25485035)
- Issuing `removeInstance()` on the last member of a cluster, and particularly the seed member, was resulting in a cluster that could not be dissolved. Now, issuing `removeInstance()` on the last member of a cluster results in an error, and you must use `dissolve()` on that instance to ensure the cluster is correctly dissolved. (Bug #25226130)
- `removeInstance()` resulted in unexpected behavior in some cases, for example when an empty password was passed as part of the URI to the instance. (Bug #25111911)
- The usage of the terms global session and active session have been made more consistent in the user interface. (Bug #24954546)
- The `removeInstance()` was not working on stopped instances and it was not possible to remove an unavailable instance from the cluster. The fix adds a new option `force` to the `removeInstance()` command to enable you to remove instances from the metadata that are permanently not available, avoiding obsolete data from being kept in the metadata of the cluster. In addition the error message provided when not using the force option has been improved and the online help for the `removeInstance()` was also updated accordingly. (Bug #24916064)
- Creating Classic sessions that connect using Unix sockets now uses the correct defaults such as hostname. This resolves the previous limitation of using Unix sockets to connect to InnoDB cluster instances. See [MySQL Shell Connections](#) for information on how the defaults are applied to socket connections. (Bug #24848763, Bug #26036466)

References: See also: Bug #24911068.

- The error messages generated by issuing `dba.deployLocalInstance()` against an unsuitable or incompatible instance have been improved. (Bug #24598272)
- The `dba.createCluster()`, `dba.getCluster()`, and `dba.rebootClusterFromCompleteOutage()` functions have been updated to validate the cluster name, using the following rules:
 - Name must start with a letter or the `_` character
 - Name can only contain alphanumeric characters and the `_` character
 - Cannot be longer than 40 characters
 - Cannot be empty

The `Cluster.addInstance()` function has been updated to validate the label used on an instance in the cluster, using the following rules:

- Label can only contain alphanumerics or the `_` character

- Cannot be longer than 256 characters
- Cannot be empty

(Bug #24565242)

- When using `cluster.rejoinInstance()` any user specified in the connection string passed in was being ignored and the connection was using the root user. (Bug #85567, Bug #25786495)
- On an instance configured as a multithreaded slave, in other words `slave_parallel_workers` set to greater than 0, and with `slave_parallel_type=DATABASE`, `dba.checkInstanceConfiguration()` was not detecting that the instance was not correctly configured for InnoDB cluster usage.
- If `removeInstance()` failed due to a connection error, an error was reported but the instance was incorrectly removed from the InnoDB cluster metadata, and remained part of the replication group. The fix ensures the metadata is correctly updated according to the result of `removeInstance()`.
- In a situation where a new primary instance was elected, adding a new instance to the cluster resulted in an error due to a failed connection to the previous primary instance.
- The functions that modify server variables, such as `dba.createCluster()` and `dba.validateInstance()` now provide more information in interactive mode output and log output about server variables which are changed when executed.
- Deploying instances to paths with directories that contained spaces was failing without error. Use double backslash to specify such paths, for example `D:\\Cluster\\foo bar`.
- The Cluster object obtained from functions such as `dba.createCluster()` or `dba.getCluster()` became unusable once the Shell session in which the object was created is was connected to a different server. The fix modifies the Cluster object so that:
 - The Cluster object holds an internal reference to the Session from which it was created or retrieved.
 - AdminAPI functions that modify the Cluster are made using the session referenced by the object.

Security Notes

- The linked OpenSSL library for MySQL Shell Commercial has been updated to version 1.0.2l. Issues fixed in the new OpenSSL version are described at <http://www.openssl.org/news/vulnerabilities.html>.

This change does not affect Oracle-produced MySQL Community builds of MySQL Shell, which use the yaSSL library instead. (Bug #26284262)

- Running queries on servers using MySQL 8 against a table with a text column caused an unexpected halt.

Functionality Added or Changed

- The `clusterAdminType` option of the `dba.createCluster()` was unused and has been removed. (Bug #25585027)
- The `--help` message now displays all short-form options.

The `--help` message now displays complete information for the `--json` option.

`shell.help('options')` now returns usable information about `mysqlsh` options.

`cluster.help('checkInstanceState')` now reports correct information.

`cluster.help('addInstance')` and `cluster.help('rejoinInstance')` now report complete information.

`help()` now reports complete information for all AdminAPI and Shell commands. (Bug #24841493, Bug #24910540, Bug #24841749, Bug #25701345, Bug #24977542, Bug #25268410, Bug #25266846)

- Calling the `modify()` or `remove()` function without a parameter caused the function to be executed against the whole collection, which could cause unexpected results such as deleting all rows in a table. To avoid this and make the behavior consistent with `update()` and `delete()`, a client-side exception is now thrown if the `modify()` or `remove()` function is called without a parameter. Now, to execute the `modify()` or `remove()` function against a collection call them with an expression that evaluates to `true`, for example `remove('true')` or `modify('true')`.

Bugs Fixed

- **X DevAPI:** Unsigned data could be incorrectly read from the database. (Bug #24912358)
- The month was incorrectly incremented on insertion of a timestamp in a table using MySQL Shell. (Bug #26423177)
- The server version is now displayed as part of the “Connected to” message. (Bug #25824753)
- Queries that used an expression of the form `collection.find(identifier IN identifier)` caused an abnormal `mysqlsh` exit. (Bug #25754078)
- If AdminAPI functions were interrupted, for example with control-C on Unix, the exit code returned by the process was 1 instead of the expected one for terminated processes, for example `SIGINT`. (Bug #25728759)
- The options in the MySQL Shell options dictionary are now fully documented. (Bug #25701345)
- The `--ssl` option has been deprecated, use the `--ssl-mode` option. Now, if you use the `--ssl` option a deprecation warning is generated and the `--ssl-mode` option is set to either `DISABLED` or `REQUIRED` based on the value used with the `--ssl` option. (Bug #25403945)
- `shell.connect()` did not report an error if an invalid argument was used. An `ArgumentError` is now issued for any invalid argument.

The following mutually exclusive pairs of options are now checked, and an error is issued if both are specified:

- `--password` and `--dbPassword`
 - `--user` and `--dbUser`
 - `--port` and `--socket`
- (Bug #25268670)

References: See also: Bug #24911173.

- The default for prompts has been made more consistent. (Bug #25039886)
- A number of issues with the output of `shell.help("prompt")` have been corrected. (Bug #25026855, Bug #25242638, Bug #25676343, Bug #25176769)
- MySQL Shell now displays an invalid year as `0000`, matching the behavior of the MySQL prompt, rather than as `0`. (Bug #24912061)
- MySQL Shell did not display fractional seconds for values in DATETIME columns. (Bug #24911885)
- The parsing of Unix sockets provided as part of a URI has been improved. (Bug #24905066)

- Some issues with the MySQL Shell command line help output were fixed. (Bug #24841749, Bug #24841493, Bug #24910540)
- If an error occurs now both the error number and error text are correctly displayed. (Bug #24816578)
- URIs were incorrectly parsed in MySQL Shell when passwords were hidden. (Bug #24793956)
- Using the `--dba enableXProtocol` option to install X Plugin was failing. (Bug #23508428)
- `mysqlsh` stopped responding if the `\source` command was given a directory (rather than file) argument. (Bug #23097932, Bug #81060)

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

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

InnoDB Cluster Bugs Fixed

- 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)

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 short form `-?` can now be used as an alias for the `--help` command-line option in MySQL Shell. (Bug #25813228)
- 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)
- 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)

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

InnoDB Cluster Bugs Fixed

- A cluster could be rebooted from a removed instance. (Bug #84766, Bug #25480022)

Functionality Added or Changed

- **X DevAPI:** 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)

- **X DevAPI:** 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 URI is now a positional argument when starting MySQL Shell at the command line. (Bug #25077429)
- 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)

- `mysqlsh` reported no error if a nonexistent Certificate Authority file was specified. (Bug #25467491)
- MySQL Shell produced no error or warning if the server certificate was expired. (Bug #25467433)
- 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)
- On Windows, a backslash character ("`\`") could not be used in 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 options, use either `--x`, `--node`, or `--classic` instead.
- The `println()` function has been added to 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 nonexistent 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.