
MySQL Connector/Node.js Release Notes

Abstract

This document contains release notes for the changes in each release of MySQL Connector/Node.js.

For additional Connector/Node.js 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: 2017-09-14 (revision: 12794)

Table of Contents

Preface and Legal Notices	1
Changes in MySQL Connector/Node.js 8.0	3
Changes in MySQL Connector/Node.js 8.0.8 (Not yet released, Development Milestone)	3
Changes in MySQL Connector/Node.js 8.0.7 (2017-07-10, Development Milestone)	3
Changes in MySQL Connector/Node.js 1.0	4
Changes in MySQL Connector/Node.js 1.0.6 (2017-03-07, Development Milestone)	4
Changes in MySQL Connector/Node.js 1.0.5 (2016-11-14, Development Milestone)	5
Changes in MySQL Connector/Node.js 1.0.4 (2016-10-10, Development Milestone)	5
Changes in MySQL Connector/Node.js 1.0.3 (2016-06-21, Development Milestone)	5
Changes in MySQL Connector/Node.js 1.0.2 (2016-04-11, Development Milestone)	5
Changes in MySQL Connector/Node.js 1.0.1 (Not released, Internal)	6
Changes in MySQL Connector/Node.js 1.0.0 (Not released, Internal)	6

Preface and Legal Notices

This document contains release notes for the changes in each release of MySQL Connector/Node.js.

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.

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 Connector/Node.js 8.0

Changes in MySQL Connector/Node.js 8.0.8 (Not yet released, Development Milestone)

Version 8.0.8 has no changelog entries, or they have not been published because the product version has not been released.

Changes in MySQL Connector/Node.js 8.0.7 (2017-07-10, Development Milestone)

MySQL Connectors and other MySQL client tools and applications now synchronize the first digit of their version number with the (highest) MySQL server version they support. This change makes it easy and intuitive to decide which client version to use for which server version.

Connector/Node.js 8.0.7 is the first release to use the new numbering. It is the successor to Connector/Node.js 1.0.6.

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

Functionality Added or Changed

- It is no longer permitted to pass an empty search condition, such as the NULL value or an empty string, to the `Collection.modify()` and `Collection.remove()` methods.
- A number of changes have been implemented for the “drop” methods:
 - The “drop” methods are now made available at the same level as the corresponding “create” methods. For example, the `dropCollection()` and `dropTable()` methods have been removed from the `XSession` class (which has now been consolidated into the `Session` class) and moved under the `Schema` class; under the same principle, the `drop()` method has been removed from the `Collection` and `Table` classes.
 - The “drop” methods now succeed even if the objects to be dropped do not exist.
 - `dropView()` is now asynchronous and behaves exactly like `dropTable()` and `dropCollection()` by implicitly executing the operation and returning a promise that will hold the result of the drop operation.
- A configuration handler interface, `mysqlx.config`, has been created for managing persisted session configurations. See [MySQL Connector/Node.js with X DevAPI](#) for details.
- There are a few changes with regard to encrypted connections to MySQL servers:
 - Connections are now encrypted by default.
 - The connection option `ssl-enable` has been replaced by the `ssl-mode` option, which has `DISABLED`, `REQUIRED` (default), and `VERIFY_CA` as its permitted values.
 - Using the `ssl-crl` option requires the use of the `ssl-ca` option and that `ssl-mode=VERIFY_CA`; this is due to an internal requirement of the Node.js core platform.
- Consolidated the `BaseSession`, `NodeSession`, and `XSession` into a single `Session` class. The following related changes were also made:
 - The `mysqlx.getNodeSession()` method is renamed to `getSession` and returns a `Session` object.
 - The `DatabaseObject.getSession()` now returns a `Session` object.

- A new client-side failover feature has been implemented: when creating a new connection, multiple hosts now can be specified in the connection string, and Connector/Node.js tries each host until a successful connection is established or until all hosts have been tried. See [Tutorial: Getting Started](#) for details
- Connector/Node.js now supports connecting to a local server using Unix sockets. See [Tutorial: Getting Started](#) for details.
- The format of the document ID value generated when adding a document to a collection has changed. It is still a string of 32 hexadecimal digits based on a UUID, but the order of digits has been changed to match the requirement of a stable ID prefix.

Bugs Fixed

- It was not possible to create a new session for a user with a SHA256 password via the PLAIN authentication mechanism. (Bug #26117627)
- The handling of large JSON arrays was problematic, and would cause an exception to be thrown. (Bug #26084604)
- Attempting to use `bind` when removing a document from a collection would not succeed, and an exception would be thrown. (Bug #26029551)
- The `Table.update()` implementation did not require a `SearchConditionStr` parameter, and not using this parameter could result in updating all the rows of a given table. A client-side exception is now thrown if the `SearchConditionStr` parameter is undefined or empty. (Bug #25993174)
- The `Table.delete()` implementation did not require a `SearchConditionStr` parameter, and not using this parameter could result in deleting all the rows of a given table. A client-side exception is now thrown if the `SearchConditionStr` parameter is undefined or empty. (Bug #25992969)

Changes in MySQL Connector/Node.js 1.0

Changes in MySQL Connector/Node.js 1.0.6 (2017-03-07, Development Milestone)

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

Functionality Added or Changed

- Added support for validating the server certificate with a given CA and/or CRL.
- Expanded support for creating TLS sessions with a URI or connection string.
- Added support for creating IPv6 sessions with a URI or connection string.
- Added support for single array or multiple argument function calls on the public API.

Bugs Fixed

- Fixed issues with `collection.bind()`. (Bug #23236379)
- Expanded support to create sessions based on unified connection strings. This also fixed parsing issues on URI and connection string corner-cases.
- Updated behavior of `collection.add([])` to avoid confusing exceptions.

Providing an empty array as an argument should always succeed, even without an active connection to the server.

Changes in MySQL Connector/Node.js 1.0.5 (2016-11-14, Development Milestone)

Functionality Added or Changed

- Added APIs for Transaction handling, which includes the `session.startTransaction()`, `session.commit()` and `session.rollback()` functions.
- Added a Table creation API.

Changes in MySQL Connector/Node.js 1.0.4 (2016-10-10, Development Milestone)

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

Functionality Added or Changed

- Changed package name from 'mysqlx' to '@mysql/xdevapi'.
- The connector can be installed into your project using Node.js's `npm` tool. Install from the download file by issuing:

```
npm install mysql-connector-nodejs-1.0.4.tar.gz
```

Or install directly from the npm repository by issuing:

```
npm install @mysql/xdevapi
```

For more information on `npm` see <http://npmjs.com>.

Bugs Fixed

- The Connector/Node.JS version number can now be retrieved from the API. For example, `"ver=require('@mysql/mysqlx/package').version"`. (Bug #24571220)
- Added the `Schema.getCollectionAsTable()` method.
- Added the `Collection.count()` and `Table.count()` methods.
- Added support for the Common URI format.
- Added **View DDL** support.

Changes in MySQL Connector/Node.js 1.0.3 (2016-06-21, Development Milestone)

Bugs Fixed

- Connector/Node.JS was unable to create a session when SSL was enabled. (Bug #23118665)

Changes in MySQL Connector/Node.js 1.0.2 (2016-04-11, Development Milestone)

MySQL Connector/Node.js is a new Node.js driver for use with the X DevAPI. This release, v1.0.2 M1, is the first development release of the MySQL Connector/Node.js 1.0 series.

The X DevAPI enables application developers to write code that combines the strengths of the relational and document models using a modern, NoSQL-like syntax that does not assume previous experience writing traditional SQL.

To learn more about how to write applications using the X DevAPI, see [X DevAPI User Guide](#). For more information about how X DevAPI is implemented in MySQL Connector/Node.js, and its usage, see <http://dev.mysql.com/doc/dev/connector-nodejs/>.

Please note that the X DevAPI requires at least MySQL Server version 5.7.12 or higher with the X Plugin enabled. For general documentation about how to get started using MySQL as a document database, see [Using MySQL as a Document Store](#).

Functionality Added or Changed

- Internal bug fixes, and released as the first developmental milestone preview release.

Changes in MySQL Connector/Node.js 1.0.1 (Not released, Internal)

Version 1.0.1 has no changelog entries, or they have not been published because the product version has not been released.

Changes in MySQL Connector/Node.js 1.0.0 (Not released, Internal)

Version 1.0.0 has no changelog entries, or they have not been published because the product version has not been released.