
MySQL Connector/J 8.0 Release Notes

Abstract

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

For additional Connector/J documentation, see [MySQL Connector/J 8.0 Developer Guide](#).

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: 2018-02-21 (revision: 14082)

Table of Contents

Preface and Legal Notices	1
Changes in MySQL Connector/J 8.0.9 (2018-01-30, Release Candidate)	3
Changes in MySQL Connector/J 8.0.8 (2017-09-28, Development Milestone)	4
Changes in MySQL Connector/J 8.0.7 (2017-07-10, Development Milestone)	7
Index	9

Preface and Legal Notices

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

Legal Notices

Copyright © 1997, 2018, 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/J 8.0.9 (2018-01-30, Release Candidate)

Version 8.0.9 Release Candidate is the first release candidate of the 8.0 branch of MySQL Connector/J, providing an insight into upcoming features. It is suitable for use with MySQL Server versions 5.5, 5.6, 5.7, and 8.0. It supports the Java Database Connectivity (JDBC) 4.2 API.

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

Functionality Added or Changed

- **X DevAPI:** In the process of refining the definition of the X DevAPI to cover the most relevant usage scenarios, the following API components have been removed from the X DevAPI implementation for Connector/J:
 - Components that support DDLs for views, including the `createView()`, `dropView()`, and `modifyView()` methods.
 - Components that support DDLs for tables, including the `createTable()`, `dropTable()`, and `modifyTable()` methods.
 - Components that support session configurations, including the `SessionConfig` object, the `PersistenceHandler` interface, the `PasswordHandler` interface, and the `SessionConfigManager` class.
- **X DevAPI:** Added the `setSavepoint()`, `rollbackTo()`, and `releaseSavepoint()` methods to the `Session` interface to support the `SAVEPOINT`, `ROLLBACK TO SAVEPOINT`, and `RELEASE SAVEPOINT` statements. See [MySQL Connector/J X DevAPI Reference](#) for more details.
- **X DevAPI:** A new `patch()` function has been added to the `ModifyStatement` interface. The function accepts an JSON-like object describing document changes and applies them to documents matched by the `modify()` filter. See [MySQL Connector/J X DevAPI Reference](#) for more details.
- **X DevAPI:** The `createIndex()` method for the `Collection` interface now has a new syntax. See [MySQL Connector/J X DevAPI Reference](#) for more details.
- **X DevAPI:** Added the following methods for single-document operations in the X DevAPI:
 - `replaceOne()`
 - `addOrReplaceOne()`
 - `getOne()`
 - `removeOne()`See [MySQL Connector/J X DevAPI Reference](#) for more details.
- **X DevAPI:** Setters and getters methods have been added for the configuration properties with the `MysqlDataSource`, `MysqlXADataSource`, and `MysqlConnectionPoolDataSource` classes.
- **X DevAPI:** The connection property `enabledTLSProtocols` can now be used to select the allowed TLS versions for an X Protocol connection to the server.

- Connector/J now supports the new `caching_sha2_password` authentication plugin, which is the default authentication plugin for MySQL 8.0.4 and later (see [Caching SHA-2 Pluggable Authentication](#) for details).

**Note**

To authenticate accounts with the `caching_sha2_password` plugin, either a [secure connection to the server using SSL](#) or an unencrypted connection that supports password exchange using an RSA key pair (enabled by setting one or both of the connecting properties `allowPublicKeyRetrieval` and `serverRSAPublicKeyFile`) must be used.

Because earlier versions of Connector/J 8.0 do not support the `caching_sha2_password` authentication plugin and therefore will not be able to connect to accounts that authenticate with the new plugin (which might include the root account created by default during a new installation of a MySQL 8.0 Server), it is highly recommended that you upgrade now to Connector/J 8.0.9, to help ensure that your applications continue to work smoothly with the latest MySQL 8.0 Server.

- Connector/J now takes advantage of the MySQL Server 8.0 data dictionary by making the connection property `useInformationSchema` true by default; this makes Connector/J, by default, access the data dictionary more efficiently by querying tables in the `INFORMATION_SCHEMA`. See [INFORMATION_SCHEMA and Data Dictionary Integration](#) for details. Users can still set `useInformationSchema` to false; but for MySQL 8.0.3 and later, some data dictionary queries might then fail, due to deprecations of older data dictionary features.
- In the past, query texts were always passed as strings to `QueryInterceptor` methods, even if the texts were not actually used by them. Now, only suppliers for the texts are passed, and the texts are only extracted by `get()` calls on the suppliers.

Bugs Fixed

- The connection property `nullNamePatternMatchesAll`, when set to false (which was the default value), caused some `DatabaseMetaData` methods to throw an error when a null search string was used with them. The behavior was not compliant with the JDBC specification, which requires that a search criterion be ignored when a null search string is used for it. The connection property has now been removed from Connector/J 8.0. (Bug #26846249, Bug #87826)
- Trying to print the query in a `PreparedStatement` using the `toString()` method after it has been closed resulted in an exception (`No operations allowed after statement closed`) being thrown. (Bug #26748909)
- When working with MySQL Server 8.0, an update or delete statement for a `CONCUR_UPDATABLE ResultSet` failed when the `ResultSet`'s primary keys included a boolean column and the character set used was not `latin1`. (Bug #26266731)
- Connector/J failed to recognize a server greeting error it received during a handshake with the server and parsed the error message as a normal greeting packet, causing an `ArrayIndexOutOfBoundsException` to be thrown. (Bug #24924097)

Changes in MySQL Connector/J 8.0.8 (2017-09-28, Development Milestone)

Version 8.0.8 Development Milestone is the latest development release of the 8.0 branch of MySQL Connector/J, providing an insight into upcoming features. It is suitable for use with MySQL Server versions 5.5, 5.6, 5.7, and 8.0. It supports the Java Database Connectivity (JDBC) 4.2 API.

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

Functionality Added or Changed

- **Packaging:** RPM and Debian packages for installing Connector/J are now available from the [Connector/J Download page](#).
- **X DevAPI:** Connector/J has implemented a new interface of the X Dev API that allows the retrieving, adding, removing, and updating of persistent session continuation data. The implementation includes the following:
 - A `SessionConfig` object that holds the information for a session configuration data set.
 - A `PersistenceHandler` interface that allows custom implementations of persistence handlers.
 - A `PasswordHandler` interface that allows custom implementations of password handling code.
 - A `SessionConfigManager` class for editing and fetching `Sessionconfig` objects, and defining instances of the `PersistenceHandler` and `PasswordHandler`.

See [MySQL Connector/J X DevAPI Reference](#) for more details.

- **X DevAPI:** A new connection property, `xdevapi.auth`, has been added for specifying the authentication mechanism for connections using the X Protocol. Allowed values are `MYSQL41`, `PLAIN`, and `EXTERNAL`. See the entry for the new property in [Configuration Properties](#) for details.
- **X DevAPI:** To support [row locks](#) for the `find()` method of the X DevAPI, the `FindStatement` and the `SelectStatement` interfaces have been extended with the following methods:
 - `lockExclusive()`, which works like `SELECT ... FOR UPDATE` for relational tables.
 - `lockShared()`, which works like the `SELECT ... LOCK IN SHARED MODE` (for MySQL 5.7) or `SELECT ... FOR SHARE` (for MySQL 8.0) for relational tables.

See [MySQL Connector/J X DevAPI Reference](#) for more details.

- **X DevAPI:** Connector/J now supports the expanded syntax for the `IN` and `NOT IN` operator, which can check if a sub-expression is contained inside another one; for example:

```
// For documents
coll.find("$.b IN [100,101,102]").execute();
coll.find("'some text with 5432' in $.a").execute();
coll.find("1 in [1, 2, 4]").execute();
coll.find("{ 'a': 3 } not in { 'a': 1, 'b': 2 }").execute();
// For relational tables
tbl.select().where("3 not in [1, 2, 4]").execute();
tbl.select().where("'qqq' not in $.a").execute();
tbl.select().where("{ 'a': 1 } in { 'a': 1, 'b': 2 }").execute();
```

- **X DevAPI:** A number of changes have been implemented for the “`drop`” methods for the X DevAPI:
 - Removed `dropCollection(schemaName, collectionName)` and `dropTable(schemaName, tableName)` from `Session`.
 - Added `dropCollection(collectionName)` and `dropTable(tableName)` to `Schema`.

- `Schema.dropView()` now executes immediately and returns `void`; also, the `ViewDrop` interface has been removed.
- `Collection.dropIndex()` now executes immediately and returns `void`; also the `DropCollectionIndexStatement` interface has been removed.
- The “drop” methods now succeed even if the objects to be dropped do not exist.
- Conversion from the MySQL `TIME` data to `java.sql.Date` is now supported. In the past, a `getDate()` retrieving data from a `TIME` column would throw an `SQLException`. Now, such a retrieval returns a `java.sql.Date` object containing the time value expressed in number of milliseconds from the Java epoch; also returned is the warning: “Date part does not exist in SQL `TIME` field, thus it is set to January 1, 1970 GMT while converting to `java.sql.Date`.” (Bug #26750807)
- A new connection property, `enabledTLSProtocols`, can now be used to override the default restrictions on the TLS versions to be used for connections, which are determined by the version of the MySQL Server that is being connected to. By providing a comma-separated list of values to this option (for example, “`TLSv1,TLSv1.1,TLSv1.2`”) users can, for example, prevent connections from using older TLS version, or allow connections to use TLS versions only supported by a user-compiled MySQL Server. See the entry for the new property in [Configuration Properties](#) for details. Thanks to Todd Farmer for contributing the code. (Bug #26646676)
- Updated the timezone mappings using the latest IANA and CLDR time zone databases. (Bug #25946965)
- A new option for the `loadBalancingStrategy` connection property called `serverAffinity` has been added. The servers listed in the new connection property `serverAffinityOrder` (which should be a subset of the servers in the host list of the connection URL) are contacted in the order they are listed until a server is available or until the list of servers is exhausted, at which point a random load-balancing strategy is used with the hosts not listed by `serverAffinityOrder`. See descriptions for `loadBalancingStrategy` and `serverAffinityOrder` in [Configuration Properties](#) for details. (Bug #20182108)

Bugs Fixed

- **Important Change:** Following the changes in MySQL Server 8.0.3, the system variables `tx_isolation` and `tx_read_only` have been replaced with `transaction_isolation` and `transaction_read_only` in the code of Connector/J. Users should update Connector/J to this latest release in order to connect to MySQL 8.0.3. They should also make the same adjustments to their own applications if they use the old variables in their codes. (Bug #26440544)
- **X DevAPI:** Calling `schema.dropView()` with a null argument resulted in a `NullPointerException`. (Bug #26750807)
- **X DevAPI:** When `dropCollection()` was applied on a null collection, a `NullPointerException` occurred. (Bug #26393132)
- When using cached server-side prepared statements, a memory leak occurred as references to opened statements were being kept while the statements were being detached; it happened when either the `close()` method has been called twice on a statement, or when there were conflicting cache entries for a statement and the older entry had not been closed and removed from the opened statement list. This fix makes sure the statements are properly closed in both cases. Thanks to Eduard Gurskiy for contributing to the fix. (Bug #26633984, Bug #87429)

- The regression test for Bug#63800 failed because the default value of the system variable `explicit_defaults_for_timestamp` of MySQL Server has been changed since release 8.0.2. The test has been adjusted to take the change into consideration. (Bug #26501245)
- Running callable statements against MySQL Server 8.0 resulted in the `SQLException: ResultSet is from UPDATE. No Data.` (Bug #26259384)
- Secure JDBC connections did not fall back to the default truststore when a custom one was not provided. (Bug #26243128)
- In `com/mysql/jdbc/ServerPreparedStatement.java`, the arguments `resultSetType` and `resultSetConcurrency` for a call of `Connection.prepareStatement()` were swapped. (Bug #25874048, Bug #85885)
- Some JDBC proxied objects were missing the proper handlings of the `equals()` methods, thus even comparison of one of these proxied objects to its own self with `equals()` yielded false. This patch introduces proper handlings for the `equals()` method in all the relevant proxies. (Bug #21931572, Bug #78313)
- A server-side prepared statement was not closed when the same statement was being prepared again while the original statement was being cached. This was caused by the silent replacement of the cache entry of the old statement by the new. When this happened repeatedly, it caused eventually the complaint that `max_prepared_stmt_count` was exceeded. This fix makes sure that when a cache entry for a statement replaces an older one, the older statement is immediately closed. (Bug #20066806, Bug #74932)

Changes in MySQL Connector/J 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/J 8.0.7 is the first release to use the new numbering. It is the successor to Connector/J 6.0.6.

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

Functionality Added or Changed

- **X DevAPI:** There are changes to some methods related to the `Result` interface:
 - `getLastDocumentId()` and `getLastDocumentIds()` have been replaced with `getDocumentId()` and `getDocumentIds()`, which are put under a new `AddResult` interface that extends `Result`.
 - A new `getAutoIncrementValue()` method is added to the new `InsertResult` interface that extends `Result`.

See [MySQL Connector/J X DevAPI Reference](#) for more details. (Bug #25207784)

- **X DevAPI:** 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.
- **X DevAPI:** Connections using the X Protocol are now secure by default. Also, the `xdevapi.ssl-enable` connection option has been replaced by the `xdevapi.ssl-mode` option, which has

`DISABLED`, `REQUIRED` (default), `VERIFY_CA`, and `VERIFY_IDENTITY` as its permitted values; see the description for the new option in [Configuration Properties](#) for details.

- **X DevAPI:** Consolidated the `BaseSession`, `NodeSession`, and `XSession` interfaces into a single `com.mysql.cj.api.xdevapi.Session` interface. The following related changes were also made:
 - Renamed `XSessionFactory` to `SessionFactory`.
 - Consolidated the `AbstractSession`, `NodeSessionImpl`, and `XSessionImpl` classes into the `com.mysql.cj.xdevapi.SessionImpl` class.
 - Removed the `Session.bindToDefaultShard()` method and the `VirtualNodeSession` interface.
 - The `mysqlx.getNodeSession()` method has been renamed to `mysqlx.getSession()` and it now returns a `Session` object.
 - The `DatabaseObject.getSession()` method now returns a `Session` object (instead of the old `Session` interface).

See [MySQL Connector/J X DevAPI Reference](#) for more details.

- To avoid using JDBC statements inside core Connector/J classes, the following changes have been implemented:
 - Created a new `com.mysql.cj.api.Query` interface, which is implemented by `StatementImpl`.
 - Replaced the `com.mysql.cj.api.jdbc.interceptors.StatementInterceptor` interface with the `com.mysql.cj.api.interceptors.QueryInterceptor` interface.
 - Added a new method, `PacketPayload preprocess(PacketPayload queryPacket)`, to `QueryInterceptor`.
 - Renamed the connection property `statementInterceptors` to `queryInterceptors`. See [Configuration Properties](#) for details.
- Added Japanese collation for the `utf8mb4` character set.

Bugs Fixed

- **X DevAPI:** `createView()` failed with a `NullPointerException` when there were null inputs to it. This fix adds checks for nulls, and makes Connector/J throw the proper errors for them. (Bug #25575156)
- **X DevAPI:** `createTable()` failed with a `NullPointerException` when there were null inputs to it. This fix adds checks for nulls, and makes Connector/J throw the proper errors for them. (Bug #25575103)
- **X DevAPI:** The connection properties `enabledSSLCipherSuites`, `clientCertificateKeyStoreUrl`, `clientCertificateKeyStoreType`, and `clientCertificateKeyStorePassword` were ignored for connections using the X Protocol. (Bug #25494338)
- **X DevAPI:** Calling `getNodeSession()` with an URL string containing SSL parameters caused a `CJCommunicationsException`. This has been fixed by creating a byte buffer to handle SSL handshake data. (Notice that `getNodeSession()` has since been consolidated into `getSession()`.) (Bug #23597281)

- **X DevAPI:** Concurrent asynchronous operations resulted in hangs, null pointer exceptions, or other unexpected exceptions. This has been fixed by correcting a number of problems with the [SerializingBufferWriter](#) and by limiting the number of buffers sent with a gathering write. (Bug #23510958)
- **X DevAPI:** When a thread failed to make a connection to the server using the X Protocol, the client application hung. A new connection property, `xdevapi.asyncResponseTimeout` (default value is 300s), now provides a duration beyond which the attempt to connect timeouts, and a proper error is then thrown. See description for the new option in [Configuration Properties](#) for details. (Bug #22972057)
- Connector/J failed a number of regression tests in the testsuite related to geographic information system (GIS) functions because of changes to GIS support by the MySQL server. The fix corrects the tests. (Bug #26239946, Bug #26140577)
- Attempts to connect to a server started with collation `utf8mb4_de_pb_0900_ai_ci` resulted in null pointer exceptions. (Bug #26090721)
- Configuration templates named by the connection property `useConfigs` were not recognized by Connector/J. (Bug #25757019, Bug #85555)
- A `NullPointerException` was returned when `getDate()`, `getTime()`, or `getTimestamp()` was called with a null `Calendar`. This fix makes Connector/J throw an `SQLException` in the case. (Bug #25650305)
- An `ArrayIndexOutOfBoundsException` was thrown when a server-side prepared statement was used and there was a `NULL` in a `BLOB`, `TEXT`, or `JSON` type column in the `ResultSet`. (Bug #25215008, Bug #84084)

Index

, 4, 7

A

authentication, 4
authentication plugin, 3

C

callable statements, 4
coalation, 7
custom load balancing, 4

D

data dictionary, 3
Debian package, 4
`dropCollection()`, 4
`dropX()`, 4

E

`enabledTLSProtocols`, 3, 4
`equals()`, 4

G

`getDate()`, 7

getTime(), 7
getTimestamp(), 7
GIS, 7

I

Important Change, 4

J

Japanese collation, 7

L

lockExclusive(), 4
lockShared(), 4

M

max_prepared_stmt_count, 4
modify(), 7

N

nullNamePatternMatchesAll, 3

P

Packaging, 4
pathc(), 3
prepared statement, 3
prepared statements, 4
preparedStatement(), 4
proxied objects, 4

Q

QueryInterceptor, 3

R

regression test, 4
regression tests, 7
remove(), 7
row locking, 4
RPM package, 4

S

SAVEPOINT, 3
schema.dropView(), 4
server greeting error, 3
server-side prepared statement, 4
serverAffinity, 4
setters and getters, 3

T

tf8mb4_de_pb_0900_ai_ci, 7
TIME, 4
time zone mappings, 4

TLS, 4
truststore, 4

U

useConfigs, 7
utf8mb4, 7

X

X DevAPI, 3, 4, 7
xdevapi.auth, 5
XSession, 7

