
MySQL Connector/J 6.0 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 6.0 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.

Document generated on: 2018-06-26 (revision: 15457)

Table of Contents

Preface and Legal Notices	1
Changes in MySQL Connector/J 6.0.6 (2017-03-10, Milestone 5)	2
Changes in MySQL Connector/J 6.0.5 (2016-10-19, Milestone 4)	4
Changes in MySQL Connector/J 6.0.4 (2016-09-06, Milestone 3)	5
Changes in MySQL Connector/J 6.0.3 (2016-06-24, Milestone 2)	6
Changes in MySQL Connector/J 6.0.2 (2016-04-11, Milestone 1)	7
Index	8

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 6.0.6 (2017-03-10, Milestone 5)

Version 6.0.6 Milestone is the fifth development release of the 6.0 branch of MySQL Connector/J, providing an insight into upcoming features. It is suitable for use with MySQL Server versions 5.5, 5.6, and 5.7. It supports the Java Database Connectivity (JDBC) 4.2 API.

- [Functionality Added or Changed](#)

- [Bugs Fixed](#)

Functionality Added or Changed

- **X DevAPI:** The `getPluginVersion()` method has been removed from Connector/J, as it is no longer supported by the X Protocol. (Bug #25056803)
- **X DevAPI:** The following new connection options have been added for SSL/TLS configuration:
 - `xdevapi.ssl-enable`
 - `xdevapi.ssl-truststore`
 - `xdevapi.ssl-verify-server-certificate`See [Configuration Properties](#) for details.
- **X DevAPI:** DDL for views are now supported by the new methods `createView()`, `dropView()`, and `alterView()`. However, the functions do not support partitioning in an InnoDB cluster or sharding.
- **X DevAPI:** The Connector/J X DevAPI has been reorganized:
 - All interfaces for public usage have been moved to the `com.mysql.cj.api.xdevapi` package, and their implementation classes to `com.mysql.cj.xdevapi`.
 - All Connector/J internal interfaces have been moved to the `com.mysql.cj.api.x.core` and `com.mysql.cj.api.x.io` packages, and their implementation classes to `com.mysql.cj.x.core` and `com.mysql.cj.x.io`.
 - Protobuf generated classes have been moved to the `com.mysql.cj.x.protobuf` package.
 - The `MysqlxSessionFactory` has been renamed `XSessionFactory`.
- IPv6 host addresses are now supported for connections using the X Protocol.
- Connector/J now supports the new character set collations implemented in MySQL Server 8.0. See [Unicode Character Sets](#) for details.

Bugs Fixed

- **X DevAPI:** The `getLastDocumentIds()` method only reported document IDs specified by users, but not those generated by Connector/J. With this fix, all IDs are now reported. (Bug #23519211)
- When Connector/J was reading a `TIMESTAMP` value into an instance of a JSR-310 data type, the reading was wrong when the time did not exist in the local time zone of the JVM due to a time change for Daylight Saving Time. It was because Connector/J created first a `java.sql.Timestamp` instance (which used the JVM's time zone) for the value and then converted it to, for example, `java.time.LocalDateTime`. With this fix, a JSR-310 object is created directly to avoid the conversion. (Bug #24658016, Bug #82964)
- Query executions using prepared statements failed with the error “Unknown ProtocolEntity class null” when the connection property `useCursorFetch` was set to `true`. (Bug #24527173)
- When server-side prepared statements were used, updates to result sets failed in errors. (Bug #24525461)
- The `getString()` method returned wrong millisecond values for the `TIMESTAMP` data type. (Bug #24512766, Bug #82707)
- Updates to a document object of the `Dbdoc` type failed when the document contained an array. (Bug #24471057)

- The method `isNumberSigned()` returned `true` for columns of type `VARCHAR`, which was wrong as it should return `false` for any non-numeric columns, according to the JDBC Specification. The behavior has now been corrected. (Bug #24350526)
- An `IllegalArgumentException` was thrown when `lenient` was false for a `Calendar` object and the hours, minutes, and seconds explicitly set via the `Calendar` constructor did not match the values resulted from the timezone conversion of the `Calendar` object's date value. (Bug #23702040, Bug #82005)
- A connection failed with `IllegalStateException: TrustManagerFactoryImpl is not initialized` after Connector/J set `javax.net.ssl.trustStore`. (Bug #23510894)
- A `NullPointerException` was thrown when a `NULL` object of any of the classes defined in the `java.time` package was retried by the `ResultSet.getObject()` method. (Bug #23188159, Bug #81202)
- The `getType()` method returned errors on columns of YEAR and DATETIME data types. (Bug #22931277)

Changes in MySQL Connector/J 6.0.5 (2016-10-19, Milestone 4)

Version 6.0.5 Milestone is the forth development release of the 6.0 branch of MySQL Connector/J, providing an insight into upcoming features. It is suitable for use with MySQL Server versions 5.5, 5.6, and 5.7. It supports the Java Database Connectivity (JDBC) 4.2 API.

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

Functionality Added or Changed

- Added support for the error codes of two MySQL server errors, `ER_XA_RBTIMEOUT` and `ER_XA_RBDEADLOCK`. (Bug #13702433, Bug #64188)
- *X DevAPI*: Client-side failover for establishing an `XSession` is now supported. See [Configuring Client-Side Failover when using the X Protocol](#) for more details.
- `com.mysql.cj.core.MySQLType` can now be used as `java.sql.SQLType` in JDBC methods.
- *X DevAPI*: A new method, `createTable()`, has been added to the `Schema` interface. See [MySQL Connector/J X DevAPI Reference](#) for details.
- *X DevAPI*: A new method, `bindToDefaultShard()`, has been added to the `XSession` interface. It creates a “virtual” `NodeSession` instance, which shares the connection to the router with the `XSession` that was used to create it. See [MySQL Connector/J X DevAPI Reference](#) for details.
- *X DevAPI*: For any function that takes a value list of parameters for its argument, there is now more flexibility with how the parameters are to be specified: they can now be supplied either as a value list or a list of individual parameters.
- The `Extension` interface has been removed. Extension classes now implement their own interfaces.

Bugs Fixed

- Connecting to a MySQL server with an invalid connection URL resulted in a `WrongArgumentException` and a stack trace. With this fix, the connection failed in the situation with an `SQLException` and without a stack trace. (Bug #24613062, Bug #82896)
- In certain cases, the exception interceptor was being triggered twice in the internal `SQLException` factory method. Also, if the exception interceptor returned an exception with the cause already

initialized, the same factory method would fail to initialize the cause again, and the real cause for the exception remained uncaptured. (Bug #23743956)

- `getLength()` returned -1 for fields of the `LONGBLOB`, `LONGTEXT`, and `GEOMETRY` data types. This is due to the way these data types are handled by protocol buffers, and this fix makes the method return the right values. (Bug #22988922)
- A memory leakage occurred when the connection properties `cachePrepStmts` and `useServerPrepStmts` were both set to be `true` and server-side prepared statements were set as non-poolable, which resulted in the prepared statement being not closable by the client, and the number of prepared statements then kept on increasing.

When the memory leakage described above occurred, it did not make Connector/J fail, as Connector/J switched to using client-side prepared statements when the maximum number of prepared statements was reached. However, when `rewriteBatchedStatements` was also set to true, the switch to client-side prepared statements did not occur, and Connector/J threw the `MySQLSyntaxErrorException` (“Can't create more than `max_prepared_stmt_count` statements”) when the client wanted to create more prepared statements than allowed.

This fix corrected the way prepared statements are handled in order to avoid both of the problems described above. (Bug #22954007, Bug #80615)

- `ResultSet.getString()` sometimes returned garbled data for columns of the JSON data type. This was because JSON data was binary encoded by MySQL using the `utf8mb4` character set, but decoded by Connector/J using the `ISO-8859-1` character set. This patch fixes the decoding for JSON data. Thanks to Dong Song Ling for contributing to the fix. (Bug #22891845, Bug #80631)
- When Connector/J retrieved the value of a `BIT` column as a string using, for example, `getString()`, it returned the wrong string if the `BIT` value happened to be equivalent to the decimal value of some ASCII character. This was because Connector/J treated the `BIT` value as a character code; thus, for example, the value “01100001” (decimal 97 in binary) was interpreted as the character “a” (whose ASCII value in decimal is 97), which was returned by the function. This fix corrects the parsing behavior of Connector/J on `BIT` values, so that a string representation of the number (“97” in the last example) is returned by `getString()`. (Bug #21938551, Bug #78685)
- When the connection property `useLocalTransactionState` was set to “true” and `autocommit` was set to “false” on the server, if any exception was thrown, any further calls for `rollback()` or `commit()` were not sent to the server. It was because when there was an exception while executing a query, Connector/J lost the information regarding the server's transaction state. This patch fixes this issue by preserving the previous transaction state for the current connection when any exception is thrown. (Bug #20212882, Bug #75209)
- An invalid connection URL caused Connector/J to throw a `NullPointerException`. With this fix, an `SQLException` is thrown instead in the situation. (Bug #18759269, Bug #72632)
- When a very large amount of compressed data is transmitted from the server to the client and under very special circumstances, a `CommunicationsException` might occur. It happened when a single compressed packet from the server was not big enough to provide an entire uncompressed packet to the client side. With this fix, Connector/J reads as many compressed packets as necessary to fill the size of the uncompressed packet that was being requested. Thanks to Ryosuke Yamazaki for contributing to the fix. (Bug #11756431, Bug #48346)

Changes in MySQL Connector/J 6.0.4 (2016-09-06, Milestone 3)

Version 6.0.4 Milestone is the third development release of the 6.0 branch of MySQL Connector/J, providing an insight into upcoming features. It is suitable for use with MySQL Server versions 5.5, 5.6, and 5.7. It supports the Java Database Connectivity (JDBC) 4.2 API.

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

Functionality Added or Changed

- Connector/J now supports the shorthand notation "*" in SQL expressions (for example, `SELECT COUNT(*) FROM mytable`) for connections using the X Protocol. (Bug #21790242)

Bugs Fixed

- On Windows platforms, SSL connections to MySQL servers using the X Protocol hung. (Bug #24301468)
- Connector/J could not be installed as an OSGi bundle, because in the Manifest of the Connector/J JAR file, the variable name for the OSGi version number was wrong. (Bug #23743947, Bug #82046)
- When Connector/J tried to read from a column of the `BIT` data type, an `ArrayIndexOutOfBoundsException` resulted. (Bug #23644816, Bug #81755, Bug #22931433)
- Connector/J failed to connect to a server when using a blank password. (Bug #21690043)

Changes in MySQL Connector/J 6.0.3 (2016-06-24, Milestone 2)

Version 6.0.3 Milestone 2 is the second development release of the 6.0 branch of MySQL Connector/J, providing an insight into upcoming features. It is suitable for use with MySQL Server versions 5.5, 5.6, and 5.7. It supports the Java Database Connectivity (JDBC) 4.2 API.

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

Functionality Added or Changed

- For MySQL server 5.7.5 and later, the EOF packet in the MySQL server/client protocol has been deprecated and replaced by the OK packet. The change is now supported by Connector/J. (Bug #23212347)
- SSL is now supported for connections to a MySQL server using the X Protocol. (Bug #21532788)
- The following objects are no longer extensions of the `Extension` interface, but get their own implementations now, which reduces their dependencies on other objects:
 - `BalanceStrategy`
 - `ProfilerEventHandler`
 - `AuthenticationPlugin`

Bugs Fixed

- An excessive amount of memory was used when the connection properties `enablePacketDebug` and `traceProtocol` were both set to "true." (Bug #23535571)
- Connector/J hung, returned a `NullPointerException`, or returned an incorrect exception when using result sets with the properties `TYPE_FORWARD_ONLY` and `CONCUR_UPDATABLE`. It was due to an inaccurate check for the cursor for the result set. This fix makes sure Connector/J checks accurately on whether a cursor has been requested, both when executing a statement and fetching its results. (Bug #23201930)
- With some Tomcat web applications, when Connector/J connects to the server with `useLegacyDatetimeCode=false` without setting `serverTimeZone`, a `NullPointerException` was returned. This was because the timezone property file for Connector/J was loaded by the bootstrap

class loader, which did not know the location of the property file and thus failed to load it. This fix avoids the problem by making Connector/J use the same class loader for both the property file and the Connector/J classes. (Bug #23197026, Bug #81214)

- When using server-side prepared statements with `profileSQL=true` and `useInformationSchema=true`, an `SQLException` (“`ResultSet is from UPDATE. No Data`”) occurred when the client tried to advance to the next row in the result set. This was due to a failure of an internal query for metadata, which is now prevented by this fix. (Bug #23188498)
- The download package for Connector/J 6.0.2 Milestone 1 contained the Developer Guide for the wrong version of Connector/J. (Bug #23111273, Bug #81089)
- A `NullPointerException` occurred in `com.mysql.cj.mysqlx.io.AsyncMessageReader` due to a race condition when there were more than 2000 concurrent connections taking place. (Bug #23044312)
- `getTimestamp()` returned wrong value for the fractional part of a `TIME` or `DATETIME` field. (Bug #22932078)
- `LoadBalanceConnectionGroupManager.removeHost()` was not removing hosts as expected. This fix tries to ensure that host removals will be successful under different situations. (Bug #22848249)

References: See also: Bug #22678872.

- For a load-balanced connection, an `ArrayIndexOutOfBoundsException` was thrown when `ConnectionGroupManager.removeHost()` was called. It was due to an error in `LoadBalancedConnectionProxy.removeHost()`, which has now been fixed. (Bug #22730682)
- A Fabric connection threw a `NullPointerException` when all hosts have been removed from its host list, or when the internal load-balanced connection became null due to inconsistency of the replication connection. This fix adds to Connector/J the abilities to handle those situations. Also, a new connection property, `loadBalanceHostRemovalGracePeriod`, has been introduced, which sets the grace period for waiting to remove the currently active host from a load-balanced connection. See the entry for the new property in [Configuration Properties](#) for details. (Bug #22678872)

References: See also: Bug #22848249.

- At every connection, Connector/J got the `sql_mode` variable from the server and tried to parse it as a number; because `sql_mode` is not a number (except for very old versions of MySQL), an `NumberFormatException` was always thrown and then caught by the code. This fix refactored the code to avoid the unnecessary throwing and catching of the error. (Bug #21181466, Bug #77171)
- The exception message in `CallableStatement()` for incorrect output parameter registration gave little detail and the wrong error code. (Bug #18068303, Bug #71131)
- On very fast servers with other third-party components accessing the data, a `ConcurrentModificationException` was sometimes thrown. This fix prevents the exception by adding a synchronization to `ConnectionImpl.closeAllOpenStatements()`, and by refactoring part of the code inside the method. (Bug #16736619, Bug #59462)
- When working with MySQL server 5.5 and 5.6, because Connector/J did not check the version number of the server it was connected to, errors and failures occurred when there was an attempt to use certain version-dependent features (for example, using fractional seconds for servers earlier than 5.6.4). With this fix, the proper exceptions are thrown in those cases.

Changes in MySQL Connector/J 6.0.2 (2016-04-11, Milestone 1)

Version 6.0.2 Milestone 1 is the first development release of the 6.0 branch of MySQL Connector/J, providing an insight to upcoming features. Although some of these are still under development, this

release includes the following new features and changes (in comparison to the current Connector/J 5.1 production release).

The major features of Connector/J 6.0 include:

- Supports MySQL 5.5, 5.6, and 5.7.
- Supports the JDBC 4.2 specification.
- Supports the Java 8 platform (use Connector/J 5.1 for Java 7 or earlier).
- Supports the new X DevAPI. 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 the [X DevAPI User Guide](#). For more information about how the X DevAPI is implemented in Connector/J, see [MySQL Connector/J X DevAPI Reference](#).

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 store, see [Using MySQL as a Document Store](#).

For other significant changes from Connector/J 5.1 to 6.0, see [Changes in Connection Properties](#) and [Changes in the Connector/J API](#).

Connector/J 6.0.2 includes all the bug fixes that have been incorporated into Connector/J 5.1.38, plus the fixes described below.

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

Functionality Added or Changed

- Removed from the Ant build target `dist` the dependency on the target `compile-testsuite`, so that the test suite is no longer compiled by default when Connector/J is being built from source. (Bug #11746728, Bug #28286)

Bugs Fixed

- `FabricMySQLDataSource.getConnection()` threw a null pointer exception when a master failover took place. (Bug #22598938)
- The OSGi manifest file in the Connector/J JAR file did not expose the MySQL Fabric packages, so the Fabric-related classes could not be resolved even though they were present in the JAR file. (Bug #22385172)

Index

C

collation, 2

G

getPluginVersion, 2

J

java.time, 2

JSR-310 data types, 2

S

SSL, 2

T

TIMESTAMP, 2
truststore, 2

V

views DDL, 2

X

X DevAPI, 2, 4

