MySQL Connector/NET Release Notes

Abstract

This document contains release notes for the changes in recent releases of MySQL Connector/NET.

For additional Connector/NET documentation, see MySQL Connector/NET 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 (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 help with using MySQL, please visit the MySQL Forums, where you can discuss your issues with other MySQL users.

Document generated on: 2025-03-20 (revision: 29837)

Table of Contents

Preface and Legal Notices

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

Legal Notices

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 for Accessibility

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/NET Version 9.x

Changes in MySQL Connector/NET 9.3.0 (Not yet released, General Availability)

Version 9.3.0 has no release notes, or they have not been published because the product version has not been released.

Changes in MySQL Connector/NET 9.2.0 (2025-01-21, General Availability)

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- Added .NET 9 GA and EF Core 9 GA support, and removed .NET 6 and EF Core 6 support. (WL #16638, WL #16654)

Bugs Fixed

- The `MySql.Data.OpenTelemetry.dll` metadata reported the version as 0.0.0.0. (Bug #37198051)
- Added `ROW_COUNT()` concurrency checking in EFCore for update operations.

  Our thanks to Ryan Kelly for the contribution. (Bug #115853, Bug #36960454)
- The `MySQLUpdateSqlGenerator` class now inherits from `UpdateAndSelectSqlGenerator` instead of `UpdateSqlGenerator`.

  Our thanks to Hikari Fujioka for the contribution. (Bug #115020, Bug #36624158)
- Added a `NULL` check to the `MySQLActivitySource` class.

  Our thanks to Maximilian Bienhüls for the contribution. (Bug #114457, Bug #36568102)
- Added a new `rewritebatchedstatements` connection option that makes the connector rewrite batched statements and execute them all at once in a single block rather than one at a time, which is default behavior.

  Note that this connection option impacts functions like `LAST_INSERT_ID()`.

  Our thanks to Sebastian Schwarzhuber for the contribution. (Bug #110553, Bug #35240186)

Changes in MySQL Connector/NET 9.1.0 (2024-10-15, General Availability)

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- Connector/NET now supports .NET 9 and Entity Framework Core 9 (preview versions). (WL #16308)
- The OpenTelemetry.Api library is no longer bundled, it now pulls in the OpenTelemetry.Api 1.9.0 NuGet package. (WL #16453)

- Added *OpenID Connect* support leveraging the new `authentication_openid_connect_client` client-side authentication plugin. *OpenID Connect* functionality is supported by MySQL Enterprise Edition Server 9.1.0 and later.

  The new `OpenidIdentityToken` connection option defines a path to a file containing the JWT formatted identity token. (WL #16491)

Bugs Fixed

- Changed EntityFramework to use the actual column name in JOIN statements to prevent clashes.

  Our thanks to Kristoffer Sjöberg for the contribution. (Bug #116028, Bug #37032982)

- Made improvements to MySqlPoolManager.GetPoolAsync related fixes that were made in Connector/NET 8.2.0.

  Our thanks to Stuart Lang for the contribution. (Bug #111759, Bug #35937318)

  References: This issue is a regression of: Bug #110717.

- Executing SELECT statements on Big-Endian client machines could cause MySQL to unexpectedly halt.

  Our thanks to Sidraya Jayagond for the contribution. (Bug #111630, Bug #35937293)

Changes in MySQL Connector/NET 9.0.0 (2024-07-01, General Availability)

- Deprecation and Removal Notes
- Functionality Added or Changed
- Bugs Fixed

Deprecation and Removal Notes

- Removed the deprecated `OldGetStringBehavior` connection option. (WL #16280)

Functionality Added or Changed

- Removed .NET Core 7 and Entity Framework Core 7 support, as they received End of Support status on May 14, 2024. (WL #16386)

Bugs Fixed

- Improved how the connection pool releases connections. (Bug #36483069)
- The `AppendReturningClause` method is now overridden to prevent an `DatabaseGeneratedOption.Computed` attribute annotation from adding an unsupported `RETURNING` clause to the generated SQL.

  Our thanks to Wang Shiyao for the contribution. (Bug #113443, Bug #36116171)

Changes in MySQL Connector/NET Version 8.x

Changes in MySQL Connector/NET 8.4.0 (2024-04-30, General Availability)

- Deprecation and Removal Notes
- Functionality Added or Changed

- Bugs Fixed

Deprecation and Removal Notes

- The deprecated `authentication_fido_client` authentication plugin and associated `FidoActionRequested` connection option was removed. Instead, use the authentication_webauthn_client authentication plugin. (WL #16150)

Functionality Added or Changed

- Expanded the metadata present in the MSI package and DLL files. (WL #16204)
- Added TLS 1.3 support for .NET Framework 4.6, and also enabled TLS 1.3 for .NET Framework 4.8 and .NET 6 and above. (WL #16176)
- Connector/NET is now ready to support the `VECTOR` data type when it becomes available with MySQL Enterprise Server. (WL #16175)
- Known limitation of this release: because the mysql_native_password authentication plugin is disabled by default as of MySQL Server 8.4.0, some unit tests may generate errors unless the plugin is enabled.

Bugs Fixed

- Setting `MinPoolSize` to a value other than 0 would throw an exception and prevent new connections. (Bug #36319784)
- Using either shared memory or a named pipe for a connection in a multithread environment would throw an exception. (Bug #36208929, Bug #36208932)
- Mapping a column to a byte array had a limit of 8000 characters, and attempting to insert a larger value would insert an empty value. (Bug #36208913)
- The pool manager could cause a deadlock to occur. Our thanks to Marek Matys for the patch. (Bug #114272, Bug #36380976)

Changes in MySQL Connector/NET 8.3.0 (2024-01-16, General Availability)

- Deprecation and Removal Notes
- Functionality Added or Changed
- Bugs Fixed

Deprecation and Removal Notes

- Calling the MySqlDataReader.GetString() method now throws an `InvalidCastException` exception if the associated column is not a string type.

  The `OldGetStringBehavior` connection option was added to allow (by setting it to `true`) restoring previous behavior by logging a deprecation warning instead of throwing the exception. This option defaults to `false`.

  This option was removed in version 9.0.0. (WL #15972)

Functionality Added or Changed

- Previously, each Entity Framework Core (EF Core) NuGet package supported a single .NET target. For instance, the connector's EF Core 8 package was able to target .NET 8.0 only. Now, the package-target support relationships are:

  - EF Core 6 targets .NET 6
  - EF Core 7 targets .NET 6, .NET 7
  - EF Core 8 targets .NET 6, .NET 7, .NET 8

  (Bug #35968775)

- Upgraded the `Google.Protobuf` dependency to version 3.25.1. (WL #16075)
- Previously, Connector/NET 8.2.0 supported .NET 8 and Entity Framework Core 8.0 (EF8) preview releases only. Connector/NET now supports all .NET 8 and EF8 releases. (WL #16035)
- Added `Microsoft.Build.Traversal` support; executing `MSBUILD dirs.proj` in the Connector/NET root folder now builds all projects in the repository (MySQL.Data, MySQL.Data.OpenTelemetry, MySQL.Web, EntityFramework, and EFCore) as defined in `dirs.proj`. (WL #15798)

Bugs Fixed

- Added an `AssemblyInfo` file for `MySQL.Data.OpenTelemtry`. (Bug #35957212)
- After an application called `Dispose()` on a connection object, it was possible to reopen the connection. Now, the connector returns an exception if an attempt is made to reopen a disposed connection. (Bug #35827809)
- The timeout period of a transaction could elapsed prior to obtaining a connection from the connection pool. This might occur because all pooled connections were neither discarded from nor returned to the pool, but remained marked as `InUse`. The fix resolves an issue that prevented the removal of a connection from the connection pool after an exception was thrown. (Bug #112123, Bug #35731216)

Changes in MySQL Connector/NET 8.2.0 (2023-10-25, General Availability)

Packaging Notes

- Updated the `Portable.BouncyCastle 1.9.0` NuGet package reference to `BouncyCastle.Cryptography 2.2.1`, which is the currently maintained package. (Bug #111361, Bug #35488339)

Functionality Added or Changed

- Combined all of the Entity Framework Core projects (6, 7, and 8) into a single project under a folder named `EFCore`. (WL #15799)
- Connector/NET now supports .NET 8.0 and Entity Framework Core 8.0 (preview versions). (WL #15792)
- Connector/NET now supports an authentication method that enables users to authenticate to MySQL Server using WebAuthn-aware devices for classic MySQL protocol connections. WebAuthn authentication is based on the FIDO and FIDO2 standards. For an overview of the supported client-side authentication plugins and authentication methods, see Connector/NET Authentication. (WL #15193)

Bugs Fixed

- Applications like Microsoft Excel and Microsoft Power BI Desktop emitted an error when used with Connector/NET 8.0.33 and later. This fix identifies and restores the missing component. (Bug #110975, Bug #35379875)
- Multiple `MySqlConnection` objects on separate threads called simultaneously could return `ArgumentException` or `InvalidOperationException` when `MySqlPoolManager.GetPoolAsync` tried to modify the shared thread-pool object concurrently from the different threads. (Bug #110717, Bug #35307501)

Changes in MySQL Connector/NET 8.1.0 (2023-07-18, General Availability)

MySQL Connector/NET 8.1.0 is a new GA release version that supersedes the 8.0 series, and is recommended for use on production systems. This release can be used against MySQL Server version 5.7 and later.

Deprecation and Removal Notes

- Previously, `mysql.proc` was used as a way of speeding up queries for procedures and parameters. Starting with MySQL 8.0, the `mysql.proc` table is no longer available and the connector now uses `information_schema`. (WL #15643, WL #15644, WL #15654)
- Support to install Connector/NET through MySQL Installer is removed. The preferred method is to install Connector/NET using NuGet packages for full functionality, or the standalone MSI file for base-level functionality (see Installing Connector/NET on Windows). (WL #15795)
- Connector/NET no longer supports the following discontinued standard and frameworks:
  - .NET Core 3.1 and 5.0
  - EF Core 3.1 and EF Core 5.0

  (WL #15797)

Packaging Notes

- Connector/NET now provides the `MySQL.Data.OpenTelemetry` NuGet package that implements `OpenTelementry.Api` and adds the `AddConectorNet` extension method to `TraceProviderBuilder`, which sets the connector as a source of application-specific telemetry that client application code emits. For details, see Enabling OpenTelemetry Tracing. (WL #15707)

X DevAPI Notes

- Application developers must now ensure the availability of the `ZstdSharp.Port` library at run time when using the noinstall packages (NuGet packages provide the dependency). This implementation replaces the bundled implementation of the `zstd` compression algorithm, which was deactivated in Connector/NET 8.0.33. (WL #15708)

Bugs Fixed

- Queries using the `MySql.Data.MySqlClient.MySqlConnection` class did not connect as expected if the connector was installed using the MSI file. This fix now adds `System.Threading.Tasks.Extensions` and `System.CompilerServices.Unsafe` DLLs to the MSI file for inclusion into the GAC. (Bug #35360846)
- Some DLL files in NuGet and No Install packages were unsigned. (Bug #35052869)
- `MySqlParameter.Clone` could return a `MySqlParameter` object with a `MySqlDbType` value that differed from the original, rather than preserving the original type. (Bug #109682, Bug #34993796)
- `MySqlParameter.MySqlDbType`, if set to `MySqlDbType.Int24`, emitted an error similar to `Out of range value for column 'data' at row 1` when a prepared statement was executed and the parameter value was a negative number. (Bug #108756, Bug #34694519)

Changes in MySQL Connector/NET 8.0.33 (2023-04-18, General Availability)

Deprecation and Removal Notes

- Data can be passed in and out of a MySQL stored procedure through the `MySqlCommand.Parameters` collection. When used with a legacy server (prior to MySQL 8.0), the command object consults the `mysql.proc` table to determine the parameters of the stored procedure. This secondary action now is deprecated and will be removed in a future release. (WL #15645)

Packaging Notes

- .NET Framework 4.6.2 replaces version 4.5.2 in target frameworks. (WL #15641)
- Connector/NET no longer provides within the install and no-install bundles any third-party libraries that the connector does not build internally. Application developers must now ensure the availability of following libraries at run time.

  For applications using OCI Authentication and SSL Certificates validation:

  - `Portable.BouncyCastle` (see https://www.nuget.org/packages/Portable.BouncyCastle)

  For applications using X DevAPI:

  - `K4os.Compression.LZ4.Streams` (see https://www.nuget.org/packages/K4os.Compression.LZ4.Streams)
  - `Google.Protobuf` (see https://www.nuget.org/packages/Google.Protobuf)

  (WL #15682)

X DevAPI Notes

- Class `MySqlX.XdevAPI.Collection<T>` is expanded to enable the use of generic types with operations performed on a collection. `Collection<T>` now includes these new methods:
  - `Add(Object[])` (replaces the `Add(T)` method)
  - `Count()`
  - `CreateIndex(String, Object)`
  - `DropIndex(String)`
  - `Find(String)`
  - `GetOne(Object)`
  - `Modify(String)`
  - `Remove(String)`
  - `RemoveOne(Object)`

  (Bug #31182148, WL #15476)

Functionality Added or Changed

- The zstd compression algorithm is deactivated in the connector. (Bug #35281610)
- Asynchronous methods used with classic MySQL protocol connections now are implemented to ensure asynchronous behavior at the level of I/O operations. Previously, some asynchronous methods were executed in a synchronous context. (Bug #70111, Bug #26477952, WL #15484)
- Connector/NET's implementation of the `authentication_oci_client` plugin now permits using a security-token file to support ephemeral key-pair authentication when integration with an external identity provider is needed for classic MySQL protocol connections. In addition, .NET applications now can set the new `OciConfigProfile` connection option (see General Options) to specify which profile in the configuration file to use for authentication. (WL #15489)

Bugs Fixed

- Queries using the `MySql.Data.MySqlClient.MySqlConnection` class did not connect as expected if the connector was installed using the MSI file. This fix now adds `System.Threading.Tasks.Extensions` and `System.CompilerServices.Unsafe` DLLs to the MSI file for inclusion into the GAC.

  

  Important

  This resolution requires Connector/NET 8.0.33.1, which supersedes the Connector/NET 8.0.33 release.

  (Bug #35360846)
- Connector/NET API member documentation was incomplete. (Bug #34975410, Bug #35141281)
- When Connector/NET inserted multiple rows using a single `MySqlCommand` object, the `MySqlCommand.LastInsertedId` property returned the ID of the first row that was inserted rather than the last row. (Bug #109683, Bug #34993798)

- A valid query formatted with tabs or new lines could return an invalid syntax error. (Bug #95226, Bug #29722378)

Changes in MySQL Connector/NET 8.0.32 (2023-01-17, General Availability)

An Important Update

- The Connector/NET `MySql.Data` version 8.0.32.1 NuGet package now is released and the newer package supersedes the original 8.0.32 package. All of the `dotnet` build warnings produced when compiling a program that references `MySql.Data` version 8.0.32 (and related issues) are resolved when referencing the `MySql.Data` 8.0.32.1 package. (Bug #109670, Bug #34990010)

Entity Framework 6 Notes

- In some cases, the function that creates SQL queries did not specify an escape character properly. This fix adds a new method to handle the cases in which a string could require an escape character and applies it before building the SQL query. (Bug #103160, Bug #34498485)
- When Connector/NET set the `@@SESSION.sql_mode` value to `'ANSI'`, it did so without considering the current value. (Bug #79678, Bug #22564126)

Pluggable Authentication

- Connector/NET 8.0.27 implemented support for the SSPI Kerberos library on Windows, which was not capable of acquiring cached credentials previously generated by using `kinit` command. Connector/NET 8.0.32 also supports GSSAPI through the MIT Kerberos library to add that capability using the `authentication_kerberos_client` authentication plugin on Windows.

  A new connection option, `KerberosAuthMode`, enables developers using the classic MySQL protocol to attempt authentication at run time using a mode value specified at design time. These mode values are permitted:

  - `AUTO` (default value) – Use SSPI and fall back to GSSAPI in case of failure.
  - `SSPI` – Use SSPI only and raise an exception in case of failure.
  - `GSSAPI` – Use GSSAPI only and raise an exception in case of failure.

  Use of the `KerberosAuthMode` connection option is restricted to Windows. Its use raises an exception in non-Windows environments. For more information about the connector's implementation of the client-side Kerberos authentication plugin, see authentication_kerberos_client. (WL #15341)

Functionality Added or Changed

- This release of Connector/NET upgrades the NuGet package reference for `Google.Protobuf` to version 3.21.9, which has been publicly reported as not vulnerable to `CVE-2022-3171`. (Bug #34859629, WL #15408)

- To enhance the performance of connections, Connector/NET now determines whether a connection string has been analyzed already before performing the analysis for a second time. (Bug #102964, Bug #32680315)
- It is now possible to pass a stream object to the `MySqlBulkLoader.Load()` method for copying bulk data to a database. Our thanks to Alexander Reinert for the patch. (Bug #74392, Bug #21049228)
- Connector/NET now fully supports .NET 7 and Entity Framework Core 7.0 (EF7) for applications that target .NET 7 exclusively. Previously, Connector/NET 8.0.31 supported the .NET 7 and EF7 previews. (WL #15334)

Bugs Fixed

- Connector/NET could not authenticate a valid client user through Oracle Cloud Infrastructure pluggable authentication. Now, the connector implements new constraints to ensure that the client-side plugin loads the `OCI.DotNetSDK.Common` assembly, which contains the required functions needed to authenticate a client user. (Bug #34851463)
- When unloading the assembly, the procedure recommended by Microsoft was not performed accurately for .NET Core frameworks. Our thanks to Gabriele Gervasi for the patch. (Bug #108837, Bug #34724334)
- An unexpected `System.NullReferenceException` message was returned when the `MySqlCommand.Cancel()` method was called for a command associated with a closed (or disposed) `MySqlConnection` object. The issue is resolved with new validation in the method that now checks the connection state. (Bug #101507, Bug #32127591)
- Some methods of the `MySqlDataReader` class that should retrieve an integer value instead could return unexpected values. This behavior had multiple causes. First, the connector was not properly informed by the `TreatTinyAsBoolean` property that it should treat type `TINYINT(1)` as `INT` or `BOOL`. In addition, Connector/NET implemented the conversion to a different type inconsistently. (Bug #99091, Bug #31087580)
- When executing multiple `INSERT` statements in one `MySQLCommand` call, the `LastInsertedId` property was set to zero because the connector read an extra `OK` packet from the server that omitted `LastInsertedId`. Now, Connector/NET avoids overriding the value if it is set. (Bug #97061, Bug #30365157)
- Improved validation now ensures that the `MySqlParameterCollection.Add()` method behaves consistently when it checks whether a named parameter has been added already. (Bug #93370, Bug #28980952)
- Values of the `Size`, `Scale`, `Precision`, `IsNullable`, and `SourceColumnNullMapping` properties were omitted during the copy operation performed by the `MySqlParameter.Clone()` method. (Bug #92734, Bug #28777779)
- The `MySqlException.Number` property for authentication-related exceptions always returned a value of zero. (Bug #78426, Bug #21830667)
- Output parameters in combination with a command of type `CommandType.Text` were not supported. This fix changed how Connector/NET manages the parameters with an output direction for text commands. (Bug #75267, Bug #20259756)

Changes in MySQL Connector/NET 8.0.31 (2022-10-11, General Availability)

- Bugs Fixed

Deprecation and Removal Notes

- The EMTrace extension is removed. The EMTrace project implements a trace-listener plugin that feeds data back to MySQL Enterprise Monitor using a REST server endpoint. (WL #15158)
- The `MySql.Data.MySqlClient.Memcached` namespace and its members now are removed. An alternative binary or text client is recommended for applications that access data using the InnoDB memcached plugin. (WL #15213)

Functionality Added or Changed

- Connector/NET now provides DNS Service (SRV) records support without depending on libraries that were previously omitted from the Connector/NET 8.0.19 (and later) NuGet packages. This update does not change how the DNS-SRV feature looks up, extracts the SRV records, or uses those records to establish a connection (see Opening a Connection Using a Single DNS Domain). (WL #14016)
- Connector/NET now supports .NET 7 (preview) and continues to support a range of frameworks (see Connector/NET Versions). (WL #15195)
- Connector/NET now fully supports EF Core 7.0 (preview) for applications that target .NET 7 exclusively. Applications that target .NET Framework (or any .NET Standard version) cannot use this feature. (WL #15194)

Bugs Fixed

- One of the links published for submitting a pull request was not well-formed. Our thanks to Adam Croot for the patch. (Bug #108290, Bug #34535732)
- NuGet was not able to display license information for Connector/NET packages accurately. Our thanks to Matthew Steeples for the patch. (Bug #108091, Bug #34477295)
- On systems running .NET 6, Connector/NET could throw an exception when trying to perform an equality check involving type `Datetime.Date`. (Bug #107618, Bug #34317220)
- After a connection timeout expired, exception type `TimeoutException` was thrown when `MySqlException` was expected. (Bug #107600, Bug #34299402)
- Incomplete dependencies prevented Connector/NET from installing and running properly. (Bug #107316, Bug #34189859)
- If an exception occurred during the execution of a SQL statement within a transaction, the subsequent transaction rollback produced a null reference exception, which could cause the Web server (IIS) process to halt unexpectedly. This fix changes the way the connection object manages rollback behavior. (Bug #107110, Bug #34107186)
- The overall performance of the `MySql.Data.MySqlClient.MySqlParameterCollection.Add(Object)` method was improved by changing the return statement. (Bug #105997, Bug #33710643)
- When a query attribute and a command parameter had the exact same name, Connector/NET overwrote the value of the attribute with the value from the parameter, which then prevented using `mysql_query_attribute_string` to retrieve an accurate value for the attribute. Connector/NET now removes all parameter names when sending a prepared statement. (Bug #105728, Bug #33620022)
- Additional validation now checks the `MySqlParameter` value assignment to ensure that binary data in a `MemoryStream` object can be read properly. (Bug #102593, Bug #32506736)

- An error was emitted after passing in `MySqlParameter.MySqlDbType` set with type `MySqlDbType.Int24` to `command.ExecuteNonQuery()` using a prepared statement. (Bug #95986, Bug #29959095)

Changes in MySQL Connector/NET 8.0.30 (2022-07-26, General Availability)

Deprecation and Removal Notes

- The `MySql.Data.MySqlClient.Memcached` namespace and its members now are deprecated. An alternative binary or text client is recommended for applications that access data using the InnoDB memcached plugin. (WL #15163)
- The EMTrace extension is deprecated. The EMTrace project implements a trace-listener plugin that feeds data back to MySQL Enterprise Monitor using a REST server endpoint. (WL #15159)

Functionality Added or Changed

- Connector/NET collation support now aligns with the collations that were either renamed or newly added by the server starting with MySQL 8.0.30. All current collations named *utf8* are renamed to *utf8mb3*.

  The new collations are:

  Norwegian (same as Danish): Norwegian collation has two codes, `nb` (Norwegian Bokmål) and `nn` (Norwegian Nynorsk)

  - utf8mb4_nb_0900_ai_ci (same as utf8mb4_da_0900_ai_ci) (*)
  - utf8mb4_nb_0900_as_cs (same as utf8mb4_da_0900_as_cs but CASE FIRST OFF) (*)
  - utf8mb4_nn_0900_ai_ci (same as utf8mb4_da_0900_ai_ci) (*)
  - utf8mb4_nn_0900_as_cs (same as utf8mb4_da_0900_as_cs but CASE FIRST OFF) (*)

  Serbian with Latin characters (same as Croatian)

  - utf8mb4_sr_latn_0900_ai_ci (same as utf8mb4_hr_0900_ai_ci)
  - utf8mb4_sr_latn_0900_as_cs (same as utf8mb4_hr_0900_as_cs)

  Bosnian with Latin characters (same as Croatian)

  - utf8mb4_bs_0900_ai_ci (same as utf8mb4_hr_0900_ai_ci)
  - utf8mb4_bs_0900_as_cs (same as utf8mb4_hr_0900_as_cs)

  Bulgarian (same as Russian)

  - utf8mb4_bg_0900_ai_ci (same as utf8mb4_ru_0900_ai_ci)
  - utf8mb4_bg_0900_as_cs (same as utf8mb4_ru_0900_as_cs)

  Galician (same as standard Spanish)

- utf8mb4_gl_0900_ai_ci (same as utf8mb4_es_0900_ai_ci)
- utf8mb4_gl_0900_as_cs (same as utf8mb4_es_0900_as_cs)

Mongolian written with Cyrillic letters (identical to Russian)

- utf8mb4_mn_cyrl_0900_ai_ci (same as utf8mb4_ru_0900_ai_ci)
- utf8mb4_mn_cyrl_0900_as_cs (same as utf8mb4_ru_0900_as_cs)

(Bug #34156197, WL #15209)

- Connector/NET now implements the `MySqlConnectionStringBuilder.TryGetValue` method. (Bug #104910, Bug #33351775)

Bugs Fixed

- **X DevAPI:** Connector/NET permitted setting a null or empty string (`""`) value to represent a document path when modifying a collection. Now, the `Set()` method enforces using the dollar sign character (`$`) as the document path argument. (Bug #34243143)
- The link to the Oracle Contributor Agreement (OCA) page specified in the `CONTRIBUTING` file was incorrect. (Bug #34082302)
- Connections to a properly configured server using the named-pipe transport protocol were not possible when running MySQL Server as a Windows service. (Bug #33974737)
- With the SSL mode set to `VerifyCA` or `VerifyFull`, SSL connections could not be established if the client certificates were chained. (Bug #33179908)
- The `MySql.Data` NuGet package was dependent on an unmaintained version of BouncyCastle. This fix replaces the dependency with Portable.BouncyCastle, which contains the BouncyCastle DLL and is current. (Bug #106370, Bug #33827732)
- The mechanism used to clean up idle connections in the connection pool performed poorly. This fix changes the idle list from type `Queue<T>` to type `LinkedList<T>` to reduce the overall number of idle connections remaining in the pool after a connection surge is over and the load requirement is reduced. (Bug #106368, Bug #33935441)
- Type `MySqlDbType.Enum` could not be used with the `MySqlParameter` method in a prepared statement. (Bug #106247, Bug #33827735)
- Previously, it was possible to cancel the `MySqlConnection.OpenAsync` operation if the `CancellationToken` object was canceled. This fix restores the original behavior. (Bug #106243, Bug #33781447)

  References: Reverted patches: Bug #28662512.
- `MySqlCommand.CommandText` emitted an error if the stored procedure to execute included one or more backtick characters (`` ` ``) in the name. (Bug #104913, Bug #33338458)

Changes in MySQL Connector/NET 8.0.29 (2022-04-26, General Availability)

X DevAPI Notes

- An exception was raised when using a `TableInsertStatement` object to insert a string value containing special characters (`- / \ % & =`). (Bug #100314, Bug #31692694)

Functionality Added or Changed

- Previously, an application could set the `SslMode=none` connection option to establish a classic MySQL protocol or X Protocol connection to a non-TLS/SSL enabled server. However, if an application attempted to establish an unencrypted connection and also submitted connection options related to TLS/SSL, Connector/NET determined that the combination of options was incompatible and raised an error. The options incompatible with `SslMode=none` are: `CertificateFile (Certificate File)`, `CertificatePassword` (`Certificate Password`), `SslCa` (`Ssl-Ca`), `SslCert` (`Ssl-Cert`), `SslKey` (`Ssl-Key`), and `tlsversion` (`tls-version`, `tls version`).

  To reduce the number of connection errors and to enhance the connection experience, Connector/NET now implements the following changes to connections:

  - The `SslMode=none` connection-option value is deprecated (marked obsolete) and is replaced with the `SslMode=disabled` connection-option value.
  - Neither `SslMode=none` nor `SslMode=disabled` raises an error when combined with other TLS/SSL connection options.
  - The restriction against submitting duplicated TLS/SSL connection options in the same connection string (or `ConnectionString` property) now is removed. Currently, .NET Framework imposes a restriction against using duplicate `MySqlSslMode` values in traditional connection strings for X Protocol connections.

  (WL #14828)

- Connector/NET now supports an authentication method that is based on the Fast Identity Online (FIDO) standard. This new authentication method permits MySQL users to establish connections to a server by interacting with a device locally, rather than by providing a password (see Connector/NET Authentication). (WL #14871)

Bugs Fixed

- The `MySqlDataReader.GetFieldValue<T>` method returned an exception for calls attempting to retrieve the value of a `BINARY` column. (Bug #33781449, Bug #106244)
- The exception type that was returned by `MySqlConnection.Open` if the call failed was changed from `MySqlException` to `AggregateException` in the Connector/NET 8.0.28 release. This fix restores the exception to the original type (`MySqlException`). (Bug #33781445, Bug #106242)
- When calling the `GetSchema("Procedure Parameters")` method of a `MySqlConnection` object, the returned table included column names without any rows representing the collection. (Bug #33674814, Bug #105181)
- When comparing a date-time LINQ expression against a database date field, Connector/NET returned an invalid operation exception due to unavailable classes that are required for an application to filter using EF Core. (Bug #32965150, Bug #103436)
- If a `SELECT` statement was formatted with new lines or tabs, then the related characters (`\n` and `\t`) were ignored during the query analysis and Connector/NET returned a syntax error. (Bug #21971751, Bug #78760)

- A primary key having a column or columns of type unsigned `BIGINT` (unsigned 64-bit integer) was not supported when the table was used with the `MySqlCommandBuilder` class to generate single-table commands. (Bug #105768, Bug #33650097)
- Although the `MySqlCommand.Clone` method returned a copy of the current command, the contents of the `MySqlCommand.Attributes` collection (`MySqlAttributeCollection` object) were not copied to the cloned command. (Bug #105730, Bug #33613687)

Changes in MySQL Connector/NET 8.0.28 (2022-01-18, General Availability)

Deprecation and Removal Notes

- The TLSv1 and TLSv1.1 connection protocols were previously deprecated in Connector/NET 8.0.26 and support for them is removed starting with this release. Connections can be made using the more-secure TLSv1.2 and TLSv1.3 protocols. (WL #14811)

Functionality Added or Changed

- Connector/NET now supports .NET 6 and continues to support a range of frameworks (see Connector/NET Versions). (WL #14734)
- Connector/NET now fully supports EF Core 6.0 for applications that target .NET 6 exclusively. Applications that target .NET Framework (or any .NET Standard version) cannot use this feature. (WL #14735)
- .NET applications now can establish connections using multifactor authentication, such that up to three passwords can be specified at connect time. The new `Password1`, `Password2`, and `Password3` connection options are available for specifying the first, second, and third multifactor authentication passwords, respectively. `Password1` is a synonym for the existing `Password` option. (WL #14653)

Bugs Fixed

- An insert query could emit a malformed communication packet error. To prevent the error, Connector/NET now omits unnecessary parameters when handling query attributes. (Bug #33380176)
- A multihost connection attempt could fail with the first host, emit an error message, and then succeed with the next host. This fix omits the initial error message to streamline the connection behavior. (Bug #30581109)
- Connector/NET did not support the `utf8mb3` character set, which could cause compatibility issues with other software components. (Bug #105516, Bug #33556024)
- When reading zero time values from `TIME NOT NULL` columns, the data reader could return a `NULL` value instead of zero, causing subsequent `GetTimeSpan()` calls to fail. Now, if Connector/NET encounters `NULL` when reading the column value, it resets the value to zero time (`'00:00:00'`). (Bug #105209, Bug #33470147)
- Microseconds specified in a time-span value were calculated incorrectly for prepared statements. (Bug #103801, Bug #32933120)
- Connector/NET now uses the C# language's implicit conversion to an enumeration when it creates a parameter with a value equal to zero. (Bug #101253, Bug #32050204)

References: This issue is a regression of: Bug #25573071.

- Attempts to execute the `GetStream` method of the `MySqlDataReader` class returned only exceptions. Now, the `GetStream(int i)` method implementation returns the expected result for `BINARY` and `GUID` type columns. (Bug #93374, Bug #28980953)
- The calculation used to determine the length of connection attribute values could produce the wrong format in MySQL login packets. (Bug #92789, Bug #28824591)
- A complete `IConvertible` implementation for the `MySqlDateTime` class was missing from the .NET provider. (Bug #82661, Bug #24495619)
- Default values being used could cause type inconsistencies when Connector/NET created a parameter. This fix modifies the `MySqlParameter` constructor to set `DbType` and `MySqlDbType` as strings. (Bug #81586, Bug #23343947)
- Batch inserts that exceeded the `max_allowed_packet` value were not processed and skipping the insert attempt did not emit an error. (Bug #80693, Bug #22913833)
- A call to `MySqlConnection::GetSchema("IndexColumns")` unexpectedly retrieved a `NULL` value and emitted an error when the database included a full-text index. (Bug #75301, Bug #20266825)
- When Connector/NET cloned a parameter, the `SourceColumnNullMapping` property was copied with the wrong value. (Bug #74533, Bug #20056757)
- When performing a batched insert using `MySqlDataAdapter`, the `MySqlCommand.GetCommandTextForBatching()` method queried `@@sql_mode` for each insert. This fix eliminates the redundant actions and modifies the behavior to use the `sql_mode` property, which is obtained at the beginning of the connection. (Bug #71626, Bug #20328929)

Changes in MySQL Connector/NET 8.0.27 (2021-10-19, General Availability)

Functionality Added or Changed

- Previously, Connector/NET implemented the `authentication_kerberos_client` plugin to support Kerberos authentication for classic MySQL protocol connections made by applications running on Linux. Support is now extended to Windows using the Security Support Provider Interface (SSPI) for classic MySQL protocol connections (see Connector/NET Authentication). (WL #14654)
- Connector/NET now supports Oracle Cloud Infrastructure pluggable authentication for classic MySQL protocol connections. The exchange of a signed token between the `authentication_oci` plugin and the client-side plugin enables .NET applications to access MySQL Database Services within a specific tenancy in a secure way without using a password. This feature is restricted to applications that target .NET Standard 2.0 (.NET Framework 4.5.*x* implementations are not supported).

  For more information about Oracle Cloud Infrastructure authentication, see Connector/NET Authentication. (WL #14708)

Bugs Fixed

- Connections made to MySQL 5.6 using sockets and the default SSL mode (`Preferred`) failed instead of providing an unsecured connection. (Bug #33191344)

- Transactions started by the `SHOW COLLATION` statement on all MySQL 8.0 server versions were not closed when the `autocommit` system variable was set to zero. The ongoing transaction generated an error. (Bug #33123597)
- Attempts to call a fully qualified procedure using the *`schema`*.*`procedure`* syntax with single quotation marks around the schema name, procedure name, or both produced an exception. (Bug #33097912)
- When querying a `CHAR(36)` column, reading a non-NULL value could emit an error if `MySqlCommand` was part of the prepared statement. (Bug #32938630, Bug #103390)

Changes in MySQL Connector/NET 8.0.26 (2021-07-20, General Availability)

Deprecation and Removal Notes

- The TLSv1 and TLSv1.1 connection protocols now are deprecated and support for them is subject to removal in a future version of Connector/NET. (WL #14554)

Functionality Added or Changed

- The `authentication_ldap_sasl_client` plugin using GSSAPI/Kerberos is disabled for .NET applications running on Windows. (Bug #32867404)
- Connector/NET now supports Entity Framework Core (EF Core) 6.0 Preview, which targets .NET 5 at the time of this release. EF Core 6.0 does not target any .NET Standard version. EF Core 6.0 will not run on .NET Framework. (WL #14578)
- Applications that use Connector/NET now can define query attribute metadata on a per-query basis, without the use of workarounds such as specially formatted comments included in query strings. This capability is implemented using the `Attributes` property of the `MySqlCommand` class:

  ```
  myCommand.Attributes.SetAttribute("queryAttribute", "value of the query attribute");
  myCommand.Attributes.SetAttribute("queryAttribute2", DateTime.Now);
  ```

  For more information, see Using MySqlCommand. (WL #14209)
- Connector/NET now implements the `authentication_kerberos_client` plugin to support Kerberos authentication for classic MySQL protocol connections made by applications running on Linux (see Connector/NET Authentication). (WL #14429)
- A new connection-string option named `DefaultAuthenticationPlugin` enables an application to specify a valid authentication plugin to use instead of the server-side default authentication plugin (see Options for Classic MySQL Protocol Only). This option applies to classic MySQL protocol connections only.

Bugs Fixed

- An error was returned during a connection attempt using either the named pipe or shared memory protocol to a MySQL server properly configured to accept these connections. (Bug #32853205)
- The `ReplaceOne` and `AddOrRplaceOne` methods in the `MySqlX.XDevAPI` namespace ignored the `_id` parameter within each implementation. This fix now validates `_id` values and raises an error when a collection's ID and document parameters are mismatched. (Bug #32763765)

- A data table declared using valid *`database`*.*`table`* syntax within an Entity Framework model could have extra database names in the generated query (for example, *`database`*.*`database`*.*`table`*). (Bug #32358174, Bug #101236)

Changes in MySQL Connector/NET 8.0.25 (2021-05-11, General Availability)

This release contains no functional changes, and is published to align its version number with that of the MySQL Server 8.0.25 release.

Changes in MySQL Connector/NET 8.0.24 (2021-04-20, General Availability)

- Deprecation and Removal Notes
- Functionality Added or Changed
- Bugs Fixed

Deprecation and Removal Notes

- The `IgnorePrepare` connection-string option was deprecated in the Connector/NET 8.0.23 release and removed in the Connector/NET 8.0.24 release.

  The removed option instructed Connector/NET to ignore all calls to `MySqlCommand.Prepare()` that were made using the classic MySQL protocol. (Bug #31872906)

Functionality Added or Changed

- Improved server disconnection handling of an X Protocol connection now creates a log entry and returns an error message, as needed, after Connector/NET receives a connection-close notice from the server. Connector/NET detects three new types of warning notices.

  Connection idle notice. This notice applies to a server connection that remains idle for longer than the relevant timeout setting. Connector/NET closes the connection when it receives the notice in an active session or while a new session is being created. An attempt to use the invalid session returns the "`Connection closed. Reason: connection idle too long`" error message.

  Server shutdown notice. If a connection-close notice is received in a session as a result of a server shutdown, Connector/NET terminates the session with the "`Connection closed. Reason: server shutdown`" error message. All other sessions that are connected to the same endpoint are removed from the pool, if connection pooling is used.

  Connection killed notice. If the connection being killed from another client session, Connector/NET closes the connection when it receives the notice in an active session or while a new session is being created. An attempt to use the invalid session returns the "`Connection closed. Reason: connection killed by a different session`" error message.

  (WL #14208)

- If a classic MySQL protocol connection experiences a server timeout, Connector/NET now reports more precise disconnection information to affected .NET applications when the server provides improved error messages. (WL #14393)

- Previously, Connector/NET added client support for the MySQL Enterprise Edition SASL LDAP authentication plugin with `SCRAM-SHA-1` and `SCRAM-SHA-256` as authentication methods. Connector/NET now also supports `GSSAPI/Kerberos` as an alternative authentication method for classic MySQL

protocol connections. SASL-based LDAP authentication does not apply to .NET applications running macOS. (WL #14210)

- The SSH Tunneling (port forwarding) feature, which was added to support MySQL products in making secure connections on Windows, is no longer needed by other products. Now, using an alternative such as Oracle Cloud Infrastructure or SSH.NET to create a tunnel is preferred. The related connection options (`SshHostName`, `SshKeyFile`, `SshPassPhrase`, `SshPassword`, `SshPort`, and `SshUserName`) are no longer valid when making Connector/NET connections, starting with this release. (WL #14562)

Bugs Fixed

- Pound symbols in JSON columns were interpreted improperly when using accent-sensitive collation. (Bug #32429236)
- Several data types could not be mapped by running `Scaffold-DbContext` on valid MySQL tables. This fix upgrades Microsoft Entity Framework libraries to the latest and also adds all previously excluded mappings to the EFCore and EFCore5 projects. (Bug #32424742, Bug #102381)
- Constructing a regular expression for each read diminished the performance of Connector/NET. This fix limits the construction to one instance, which now is reused. (Bug #32386454, Bug #101714)
- Incomplete GUID mapping in the Entity Framework Core implementation caused an error when the `Contains` method was used to filter records. (Bug #32173133, Bug #93398)
- Additional error codes now prevent unexpected exceptions after a query. Thanks to Stanislav Revin for the patch. (Bug #32150115, Bug #101592)
- An exception was thrown if any `CHAR(36)` columns containing a `NULL` value were referenced in a query. New validation now checks for `NULL` values when the `MySqlDbType` member is `Guid`. (Bug #32049837, Bug #101252)

Changes in MySQL Connector/NET 8.0.23 (2021-01-18, General Availability)

Deprecation and Removal Notes

- The `IgnorePrepare` connection-string option was deprecated in the Connector/NET 8.0.23 release and removed in the Connector/NET 8.0.24 release.

  The removed option instructed Connector/NET to ignore all calls to `MySqlCommand.Prepare()` that were made using the classic MySQL protocol. (Bug #31872906)

- The following synonyms for the `Server` connection string option were deprecated in Connector/NET 8.0.22 and removed in 8.0.23: `address`, `addr`, and `network address`. (Bug #31248601)

Functionality Added or Changed

- Connector/NET updates to SSH ciphers and algorithms are:
  - Encryptions: `aes192-cbc`, `aes256-cbc` (deprecated); `3des-cbc`, `blowfish-cbc`, `twofish-cbc`, `twofish128-cbc`, `twofish192-cbc`, `twofish256-cbc` (invalid)

  - Host Key Algorithms: `ssh-rsa` (deprecated); `sh-dss` (invalid)
  - Key Exchange Algorithms: `diffie-hellman-group-exchange-sha1` (invalid)
  - Keyed Hash Message Authentication Codes: `hmac-ripemd160`, `hmac-ripemd160@openssh.com`, `hmac-sha1-96` (invalid)

  (Bug #31917057)

- ASP.NET applications using the MySQL provider model (`MySql.Web`) can now target .NET Framework 4.8. (Bug #31799902)
- Previously, Connector/NET added client support for the MySQL Enterprise Edition SASL LDAP authentication plugin with `SCRAM-SHA-1` as an authentication method. Connector/NET now also supports `SCRAM-SHA-256` as an alternative authentication method for classic MySQL protocol connections. `SCRAM-SHA-256` is similar to `SCRAM-SHA-1` but is more secure. SASL-based LDAP authentication does not apply to .NET applications running macOS. (WL #14255)
- With the availability of Entity Framework Core 5.0 in addition to Entity Framework Core 3.1, Connector/NET now provides two distinct EF Core NuGet packages. The split enables the connector to support both feature sets as they diverge. Initially, Connector/NET supports a partial EF Core 5.0 feature set (equivalent to EF Core 3.1) in this release. For more information, see Entity Framework Core Support.

  Breaking change: all `MySql.Data.EntityFrameworkCore.xxx` namespaces are renamed to `MySql.EntityFrameworkCore.xxx`.

  New EF Core package naming for Connector/NET 8.0.23:

  - 5.0.0+m8.0.23
  - 3.1.10+m8.0.23

  (WL #14214)

Bugs Fixed

- If a prepared statement had no parameters, Connector/NET included in the `COM_STMT_EXECUTE` packet structure a byte corresponding to `new-params-bound-flag` instead of sending the byte only when the number of parameters was greater than zero. (Bug #32208427)
- Incomplete validation limited the expected range of values that a stored procedure with a parameter of type `Boolean` could assign using the `MySqlParameter.MySqlDbType` property. (Bug #32066024, Bug #101302)
- Stronger validation was applied to information contained in the certificate store for connections made using `SslMode`. (Bug #31954655)
- A connection timeout was added to prevent the `MySqlConnection.Open` method from waiting indefinitely for a response after MySQL Router restarted unexpectedly. (Bug #31945397, Bug #100692)
- A cast made with the wrong data type during a valid EF Core operation returned an exception. (Bug #31860492, Bug #100773)
- Connector/NET used the value of -1 internally to ensure that a parameter without an index was added to the end of the parameter list. However, if an index with an actual value of -1 was passed in, the collection was interpreted as having no index and the argument did not generate an out-of-range exception. (Bug #31754599, Bug #100522)

- Without validation, an underlying 64-bit enumeration value passed in as a MySQL command parameter defaulted to type `Int32` and produced an overflow exception. (Bug #25467610, Bug #84701)

Changes in MySQL Connector/NET 8.0.22 (2020-10-19, General Availability)

Deprecation and Removal Notes

- The following synonyms for the `Server` connection string option were deprecated in Connector/NET 8.0.22 and removed in 8.0.23: `address`, `addr`, and `network address`. (Bug #31248601)

Functionality Added or Changed

- Revisions to the `MySql.Data.EntityFrameworkCore` namespace reduced the number of public classes. (Bug #31353208)
- Previously, the client-side `mysql_clear_password` authentication plugin was not supported. Now, it is permitted to send passwords without hashing or encryption by using `mysql_clear_password` on the client side together with any server-side plugin that needs a clear text password, such as for LDAP pluggable authentication. Connector/NET returns an error if the `mysql_clear_password` plugin is requested, but the connection is neither encrypted nor using Unix domain sockets. For usage information, see Client-Side Cleartext Pluggable Authentication. (Bug #30340510, WL #14002)
- For enhanced security of the existing `AllowLoadLocalInfile` connection string option, a single folder that is safe to upload files from now can be specified with the new `AllowLoadLocalInfileInPath` option (see Options for Classic MySQL Protocol Only). (WL #14093)
- Connector/NET now supports Entity Framework 6.4, which extends the compatibility of the provider to include the Linux and macOS platforms when used with the Universal Windows Platform (UWP) .NET implementation (see Entity Framework 6 Support). Connector/NET continues to support the .NET Framework implementation of Entity Framework. (WL #14076)
- Connections made using the MySQL Enterprise Edition SASL LDAP authentication plugin now are supported on Windows and Linux, but not on macOS. Connector/NET implements the `SCRAM-SHA-1` authentication method of the SASL authentication protocol. (WL #14116)
- The new `compression-algorithms` connection option sets the order by which supported algorithms are negotiated and selected to send compressed data over X Protocol connections (see Options for X Protocol Only). (WL #14001)
- In addition to providing continued support for .NET Core and .NET Framework, Connector/NET now includes support for the new .NET 5.0 framework. Compatibility testing was performed with the preview versions of .NET 5.0 and Visual Studio to encourage the efforts of early adopters. Among other capabilities, .NET 5.0 offers uniform runtime behaviors and developer experiences by taking the best of .NET Core, .NET Framework, Xamarin, and Mono. (WL #14044)

Bugs Fixed

- Procedure names were malformed before being sent to the server when the database name was not specified in the connection string. (Bug #31669587, Bug #100306)
- Microseconds were deserialized incorrectly when `MySqlCommand.Prepare()` was called for a statement that selects a `TIME(n)` column, resulting in a loss of trailing zeros in the returned result.

Now, the `MySqlTime` class calculates ticks, rather than converting the microseconds to a string. (Bug #31623730, Bug #100218)

- `MySqlConnection.GetSchema("Procedures")` returned the literal string `System.Byte[]` as the value of the `ROUTINE_DEFINITION` column, rather than the actual routine definition. (Bug #31622907, Bug #100208)
- Valid query parameters of type `DateTime` were misinterpreted as a string values. (Bug #31598178, Bug #100159)
- A mismatch of data types between the parameter of a stored procedure and the corresponding `MySqlParameter` when the `Prepare()` method was called did not generate an exception. (Bug #31458774, Bug #99793)
- An SQL syntax error was reported on valid code for creating a dynamic dropdown list from data in an ASP.NET Core application. (Bug #31337609, Bug #99523)
- Entity Framework code-first migration omitted the schema attribute that was assigned to an entity, although the automatic migration appeared to generate a table with the proper schema value at first. Subsequent queries using the schema name returned errors. (Bug #31323788, Bug #94343)
- The mapping from the `TINYINT` and `BIT` data types to `BOOLEAN` was not performed as expected when scaffolding was used. (Bug #31304070, Bug #99419)
- The Entity Framework Core migration script replaced `NULL` with `NOT NULL` on a column in the migrated table when the `MaxLength` attribute of the property for it was changed in the model. (Bug #31070175, Bug #96913)
- The `Ubiety.Dns.Core.dll` binary included with the `MySQL.Data` NuGet package was built in debug mode, which prevented the publishing of applications to Microsoft Store. (Bug #31061034, Bug #98955)
- An application using Entity Framework code-first migration without the default system decimal separator, the period character (`.`), could not generate a new database. Now, setting the system decimal separator to a different character is permitted. (Bug #30965702, Bug #94358)
- Every column of type `CHAR(36)` was interpreted as a `GUID`, which could cause the first query made by a restarted application using the `MySQL.Data` package to return an exception. This fix introduces a dedicated format (`8-4-4-4-12`) to interpret the column type properly. (Bug #29963760, Bug #93399)
- Connector/NET code did not read from the `MySql.Data.Properties.ReservedWords.txt` assembly resource consistently and could return an incomplete set of reserved keywords. (Bug #27536342, Bug #89639)
- No error occurred when `MySqlCommand.CommandTimeout` was set to a negative number, however, subsequently setting it to a positive number returned an exception. (Bug #26574860, Bug #87316)
- The `TcpClient` implementation limited some of the connection options when an external wrapper from a Windows Forms application made the connection. This fix enables the related external destructor to be called without returning an error. (Bug #26427802, Bug #82810)
- An exception was returned in debug mode after a command was canceled within a connection that was not null when `CancelQuery` was called. Thanks to Denis Yarkovoy for the improved validation patch. (Bug #26362494, Bug #86836)
- The `DbContext.Database.Migrate()` method did not succeed because the `__efmigrationshistory` table was not found. This fix modifies the method that validates the existence of the table. (Bug #25901276, Bug #85902)

- When a parameter value of zero was passed to the `MySqlParameter` constructor, Connector/NET used the `MySqlDbType` enumeration by default for type mapping and changed the value to `NULL`. This fix maps the default type to `int32`. (Bug #25573071, Bug #85027)
- Entity Framework code-first migration excluded the length specifier in the resulting binary columns. (Bug #23171349, Bug #81179)
- `Create Table` statements generated with Entity Framework were missing some of the semicolons, which caused MySQL Server to return errors. (Bug #22669961, Bug #80159)

Changes in MySQL Connector/NET 8.0.21 (2020-07-13, General Availability)

Deprecation and Removal Notes

- The following ciphers and algorithms are deprecated for SSH connections made using Connector/NET:

  Encryptions

  - 3des-cbc

  Key Exchange Algorithms

  - diffie-hellman-group14-sha1
  - diffie-hellman-group-exchange-sha1

  Keyed Hash Message Authentication Codes

  - hmac-ripemd160
  - hmac-sha1
  - hmac-sha1-96

  (Bug #31030347)

Bugs Fixed

- Connector/NET returned an error when the name of a database or stored procedure contained one or more period characters. Now, names with this format can be used when the name is enclosed properly between grave accent (`) symbols; for example, `` `db_1.2.3.45678` ``. (Bug #31237338, Bug #99371)
- An error was generated when the database name within a connection string that was passed to MySQL 5.6 or MySQL 5.7 did not match the casing used to search a related stored procedure. (Bug #31173265)
- In Connector/NET 8.0.19, calling `new MySqlConnection(null)` returned `NullReferenceException`, rather than returning an object with a `ConnectionString` property equal to `String.Empty` as the previous versions of Connector/NET did. This fix restores the earlier behavior. (Bug #30791289, Bug #98322)
- An expected empty result set generated by executing `MySQLDataReader` for a stored procedure instead returned a data table containing the `@_cnet_param_value` column. This fix eliminates an internal error that affected the result set and now `GetSchemaTable()` returns a `null` value as expected. (Bug #30444429, Bug #97300)

- The `BLOB` type was inferred internally when a value or object of type `MySqlGeometry` was used in different situations, which caused to server to return either zero matching rows or an exception. (Bug #30169716, Bug #96499, Bug #30169715, Bug #96498)
- Attempts to execute a function or stored procedure returned a `null` exception unexpectedly when the caller was not the creator of the routine. This fix introduces a mechanism to manage `null` values for these cases, permits the granting of privilege to `SHOW_ROUTINE`, and revises `SqlNullValueException` to identify when a user account lacks adequate permission to access a routine. (Bug #30029732, Bug #96143)
- Columns of type `BIGINT` in a table that was loaded using `MySqlDataReader` did not include the `UNSIGNED` flag, even though `UNSIGNED` was specified in the `CREATE TABLE` statement. An exception was generated if the value of such a column exceeded 2147483647. (Bug #29802379, Bug #95382)
- The microseconds value in the return results was set to zero consistently when `SqlCommand.Prepare()` was called for a `SELECT` statement with a `TIME(n)` column. This fix revises the way the value is produced to ensure accurate results. (Bug #28393733, Bug #91770)
- The isolation level set for a transaction did not revert to using the session value after the transaction finished. (Bug #26035791, Bug #86263)
- A valid call made to the `MySqlSimpleRoleProvider.AddUsersToRoles` method failed to execute because it violated the foreign key constraint. This fix removes an error from the code that gets the role ID. Thanks to Stein Setvik for the patch. (Bug #25046352, Bug #83657)
- The absence of a target schema in the generated `WHERE` clause of a query produced during an Entity Framework migration caused an error when the identical table was present in multiple databases. This fix adds the `table_schema` column to the generated SQL query. (Bug #23291095, Bug #72424)

Changes in MySQL Connector/NET 8.0.20 (2020-04-27, General Availability)

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- Connector/NET now supports Entity Framework Core 3.1.1 on all platforms with NET Standard 2.0 support. Microsoft Entity Framework Core 3.1.1 is not compatible with the previous versions of EF Core and those previous versions (2.1, 2.0, 1.1) are not supported by this release of Connector/NET (see Entity Framework Core Support).

  In addition, the `MySql.Data.EntityFrameworkCore.Design` NuGet package is deprecated and the functionality provided by that package now is merged with the `MySql.Data.EntityFrameworkCore` package. (WL #13793)
- Connector/NET now provides compression of X Protocol payload data, which can be configured using the new `Compression` connection option. The option is set to `preferred` mode by default to compress the payload data if the MySQL server instance also supports compression. For a description of each option value, see Options for X Protocol Only. (WL #12980)
- *Document Store:* Connector/NET now provides JSON schema validation for a collection to enforce a certain structure that documents must adhere to before they are permitted to be inserted or updated. Schema validation is performed by the server, which returns an error message if a document in a collection does not match the schema definition or if the server does not support validation.

  The existing `Schema.CreateCollection` method now is overloaded and can be used to pass a `CreateCollectionOptions` object with a schema definition to a MySQL server.

The `ReuseExistingObject` parameter of the original method is set as an option within `CreateCollectionOptions` when using the new overloaded method. The level of enforcement (`off` or `strict`, `strict` by default) and schema definition are specified using the `validation` option, for example:

```
var collOptions = CreateCollectionOptions() {
  reuseExistingObject = false,
  validation = Validation() {
    level = ValidationLevel.Strict,
    schema = "{\"id\": \"http://json-schema.org/geo\","
             + "\"$schema\": \"http://json-schema.org/draft-06/schema#\","
             + "       \"description\": \"A geographical coordinate\","
             + "       \"type\": \"object\","
             + "       \"properties\": {"
             + "          \"latitude\": {"
             + "             \"type\": \"number\""
             + "          },"
             + "          \"longitude\": {"
             + "             \"type\": \"number\""
             + "          }"
             + "       },"
             + "       \"required\": [\"latitude\", \"longitude\"]"
             + "  }"
    }
};

var coll = schema.CreateCollection("longlang", collOptions);
```

In addition, a new method permits the schema validation of an existing collection to be reset. The `Schema.ModifyCollection` method passes a `ModifyCollectionOptions` object to the server. The `validation` collection option must include either a modified `level` value or `schema` value (or both), for example:

```
var collOptions = ModifyCollectionOptions() {
  validation = Validation() {
    level = ValidationLevel.Off
      }
};

var coll = schema.ModifyCollection("longlang", collOptions);
```

The `ReuseExistingObject` option is not supported for modifications and returns an error message if it is used. (WL #13007)

Bugs Fixed

- A connection made to a named server with multiple DNS entries pointing to different IP addresses for the same server generated an exception. Now, only the first element is returned when multiple elements are found. (Bug #30970949, Bug #97448)
- The `MySQL.Data` NuGet package for Connector/NET 8.0.19 included an unsigned version of `Ubiety.Dns.Core.dll`, which produced an exception when loaded. (Bug #30798305, Bug #98204)
- Scaffolding a MySQL database with EF Core 3.0 was not implemented by Connector/NET and the connector returned an exception in response to its use. Support for EF Core 3.1.1 in this release adds scaffolding capabilities. (Bug #30677382, Bug #98011)
- The `get_info` method was not included in any of the Entity Framework Core versions (1.1, 2.0, and 2.1) that Connector/NET supported. Connector/NET now supports EF Core 3.1.1 and implements the `MySql.Data.EntityFrameworkCore.Infrastructure.MySQLOptionsExtension.Internal.get_in` method. (Bug #30347893, Bug #96990)

- The `MySqlDbType.JSON` type when used as a parameter in a prepared statement produced code errors. Connector/NET now interprets `MySqlDbType.JSON` as `MySqlDbType.VarChar`. No code changes are required to specify a JSON column. (Bug #29959124, Bug #95984)
- Blank spaces mixed with values in the `IN()` list of a `SELECT` statement generated an error. (Bug #29838254)
- An attempt to read the record of a model class defined to correspond to a MySQL table with a property of type `bool?` (nullable Boolean), using the EF Core database context, returned an error message. (Bug #29833103, Bug #93028)
- Access to the `MySqlDataReader` object was restricted when the parent `MySqlCommand` object was closed. This fix modifies `MySqlCommand.Dispose()` to no longer call the `ResetReader` method. (Bug #27441433, Bug #89159)

Changes in MySQL Connector/NET 8.0.19 (2020-01-13, General Availability)

Functionality Added or Changed

- Connector/NET supports TLS protocol versions TLSv1, TLSv1.1, TLSv1.2, and TLSv1.3. A new connection-string option, `tlsversion`, permits the restriction of a connection to a single version or to a list with any combination of the four supported TLS versions (see Options for Both Classic MySQL Protocol and X Protocol).

  Known issue: Both .NET Core 3.0 (cross platform) and .NET Framework 4.8 (windows only) added support for TLSv1.3. Be sure to confirm that the platform operating system running your application also supports TLSv1.3 before using it exclusively for connections. (Bug #30225427, WL #12748)

- Support for DNS Service (SRV) records now provides an alternative to specifying individual hosts in the connection string. Instead, a single DNS domain can map to multiple targets (servers) using SRV address records. Each SRV record includes the host name, port, priority, and weight. For .NET applications using X Protocol, a new URI scheme of `mysqlx+srv://` enables connections to share the query load when a single DNS domain is mapped to multiple servers (see Connections Using DNS SRV Records).

  Similarly, the new `dns-srv` connection-string option also enables DNS SRV lookups for connections using either the classic MySQL protocol or X Protocol. The DNS SRV feature is disabled by default. For usage information, see Options for Both Classic MySQL Protocol and X Protocol.

  Known Issue: The `MySql.Data.dll` package from the NuGet gallery is missing libraries needed by .NET Framework projects (.NET Core projects are not affected). To enable this feature, download the no-install version of MySQL Connector/NET (`mysql-connector-net-8.0.19.msi`) from https://dev.mysql.com/downloads/connector/net/ and then add `v4.5.2\MySql.Data.dll` as a reference to your project. No other references are required if all items remain in the same location. (WL #13368)

- When creating a new connection using classic MySQL protocol, multiple hosts can be tried until a successful connection is established. A list of hosts can be given in a connection string, with or without priorities.

```
// Example with priority
server=(address=192.10.1.52:3305,priority=60),(address=localhost:3306,priority=100);

// Example without priority and with multiple ports
host=10.10.10.10:3306,192.101.10.2:3305,localhost:3306;uid=test;password=xxxx;
```

If the priority is not included, or if multiple hosts have the same priority, Connector/NET selects a host at random. The same random selection behavior also applies to connections made using X Protocol, which previously selected hosts sequentially when no priority was specified. (WL #13304)

Bugs Fixed

- Clone connections did not process all connection settings as expected. (Bug #30502718)
- Connector/NET files displayed an unlikely date after the NuGet package containing them was installed in a project. (Bug #30471336, Bug #97390)
- The inclusion of the `System.Resources.Extensions` dependency was transient and now is removed from the `MySql.Data` NuGet package. (Bug #30421657, Bug #97218)

Changes in MySQL Connector/NET 8.0.18 (2019-10-14, General Availability)

Functionality Added or Changed

- Connector/NET now supports IPV6 connections made using the classic MySQL protocol when the operating system on the server host also supports IPV6. (Bug #29682333)
- Support for .NET Core 3.0 was added. (WL #13114)
- In tandem with Microsoft, Connector/NET ends support for .NET Core 1.0 and 1.1 (and also for Entity Framework Core 1.1, which depends on .NET Core 1.1). (WL #13387)
- Previously, if the server restricted a classic Connector/NET session to sandbox mode and the password on the account expired, the session continued to permit the use of `SET` statements. Now, `SET` statements in sandbox mode with an expired password are prohibited and will return an error message if used. The one exception is `SET PASSWORD`, which is still permitted (see Server Handling of Expired Passwords). (WL #13305)

Bugs Fixed

- The `Renci.SshNet.dll` deployment was problematic for Connector/NET 8.0.17 MSI installations. Some applications, such as Microsoft Excel, were unable to read MySQL data as a result. This fix removes unnecessary dependencies on the DLL and also ensures that the MSI installation deploys the correct `Renci.SshNet.dll` to the GAC. (Bug #30215984, Bug #96614)
- Connector/NET returned an inaccurate value for the `YEAR` type when a prepared command was used. (Bug #28383721, Bug #91751)
- *Entity Framework Core:* A syntax error was generated during an operation attempting to rename a table that was previously migrated from code. Now, the primary key constraint for an existing table can be dropped without errors when the follow-on migration operation is performed. (Bug #28107555, Bug #90958)

Changes in MySQL Connector/NET 8.0.17 (2019-07-22, General Availability)

Deprecation and Removal Notes

- *Document Store:* The `Where()` method is deprecated (`Obsolete` attribute applied) and will return a warning when called in the following method constructs: `Find().Where()`, `Modify().Where()`, and `Remove().Where()`. (WL #12983)

Functionality Added or Changed

- Connector/NET supports SSL PEM certificate versions 1, 2, and 3 to enable use with the full range of applications that generate certificates. (Bug #29756058)
- Support was added for .NET Core 2.2, which is a cross-platform version of .NET for building applications that run on Linux, macOS and Windows (see Connector/NET Versions). (WL #12337)
- New `README.md` and `CONTRIBUTING.md` files now accompany MySQL Connector/NET code for compatibility with Git. Distribution packages (NuGet, MSI, ZIP) continue to include the original `README` file, but do not include the new files. (WL #12970)
- Connector/NET now supports the new `utf8mb4_0900_bin` collation added for the `utf8mb4` Unicode character set in MySQL 8.0.17. For more information about this collation, see Unicode Character Sets. (WL #13099)
- *Document Store:* Connector/NET now supports the `OVERLAPS` and `NOT OVERLAPS` operators for expressions on JSON arrays or objects:

  ```
  expr OVERLAPS expr
  expr NOT OVERLAPS expr
  ```

  Suppose that a collection has these contents:

  ```
  [{
     "_id": "1",
     "list": [1, 4]
   }, {
     "_id": "2",
     "list": [4, 7]
  }]
  ```

  This operation:

  ```
  var res = collection.Find("[1, 2, 3] OVERLAPS $.list").Fields("_id").Execute();
  res.FetchAll();
  ```

  Should return:

  ```
  [{ "_id": "1" }]
  ```

  This operation:

  ```
  var res = collection.Find("$.list OVERLAPS [4]").Fields("_id").Execute();
  res.FetchAll();
  ```

  Should return:

  ```
  [{ "_id": "1" }, { "_id": "2" }]
  ```

  An error occurs if an application uses either operator and the server does not support it. (WL #12749)

- *Document Store:* For index specifications passed to the `Collection.CreateIndex()` method, Connector/NET now supports indexing array fields. For example, consider a collection with this array:

```
Session session = MySQLX.GetSession(connString);
Schema schema = session.GetSchema(schemaName);
Collection coll = schema.CreateCollection(collectionName);

var docs = new[]
{
  new {  _id = 1, name = "John Smith", emails = [ "john.smith@mycompany.com", "jsmith@php.net", "jsmith@
};

coll.Add(docs).Execute();
```

A single index field description can contain a new member name `array` that takes a `Boolean` value. If set to `true`, the field is assumed to contain arrays of elements of the given type. In addition, the set of possible index field data types (used as values of member `type` in index field descriptions) is extended with type `CHAR(N)`, where the length `N` is mandatory. For example, to create the emails_idx index with an array field:

```
coll.CreateIndex("emails_idx",
    "{\"fields\": [{\"field\": $.emails,
                    \"type\":\"CHAR(128)\",
                    \"array\": true }]}"
);
```

To find an element of the array:

```
collection
   .Find(":mail IN $.emails")
   .Bind("mail", "jsmith@php.net")
   .Execute();
```

(WL #12176)

- New support for SSH tunneling enables Connector/NET to create secure connections to a remote MySQL server using TCP/IP over SSH. With SSH server authorization, an application can establish a connection from behind a firewall when the MySQL Server port is blocked. The new connection-string options (and equivalent class properties) for SSH tunneling are supported by both the classic MySQL protocol and X Protocol connections. (WL #12747)

Bugs Fixed

- The BouncyCastle assembly was loaded into memory whenever a connection attempt was made using any SSL mode type, except `None`. Now the assembly loads only when the SSL mode type is `VerifyCA` or `VerifyFull`, or when PEM certificates are used. (Bug #29611216)
- *Document Store:* The `MySqlConnection.GetSchema()` method sometimes returned columns in an unexpected order when used with the `INFORMATION_SCHEMA.COLUMNS` table. This fix ensures that returned columns now correspond to the ordinal position only. (Bug #29536344)
- The `InvariantCulture` property was missing from some data types, which created issues during platform migration operations. Thanks to Effy Teva for the patch. (Bug #29262195, Bug #94045)
- Connector/NET connections executed `SHOW VARIABLES` unnecessarily. (Bug #28928543, Bug #93202)
- Connector/NET access to MySQL stopped working after the computer hosting the server was started and continued to operate uninterrupted for a defined period of time. (Bug #26930306, Bug #75604)

Changes in MySQL Connector/NET 8.0.16 (2019-04-25, General Availability)

- Functionality Added or Changed

- Bugs Fixed

Functionality Added or Changed

- *Document Store:* Support was added for the `->` operator to be used with JSON document paths in relational statements. For example:

  ```
  table.Select().Where("additionalinfo->$.hobbies = 'Reading'");
  ```

  (Bug #29347028)

- *Document Store:* The performance for statements that are executed repeatedly (two or more times) is improved by using server-side prepared statements for the second and subsequent executions. This happens internally; applications need take no action and API behavior should be the same as previously. For statements that change, repreparation occurs as needed. Providing different data values or different `OFFSET` or `LIMIT` clause values does not count as a change. Instead, the new values are passed to a new invocation of the previously prepared statement. (WL #12174)

- *Document Store:* Connector/NET now supports the ability to send connection attributes (key-value pairs that application programs can pass to the server at connect time). Connector/NET defines a default set of attributes, which can be disabled or enabled. In addition, applications can specify attributes to be passed together with the default attributes. The default behavior is to send the default attribute set.

  The aggregate size of connection attribute data sent by a client is limited by the value of the `performance_schema_session_connect_attrs_size` server variable. The total size of the data package should be less than the value of the server variable. For X Protocol applications, specify connection attributes as a `connection-attributes` parameter in a connection string. For usage information, see Options for X Protocol Only.

  For general information about connection attributes, see Performance Schema Connection Attribute Tables. (WL #12514)

- *Document Store:* Connector/NET now has improved support for resetting sessions in connection pools. Returning a session to the pool drops session-related objects such as temporary tables, session variables, and transactions, but the connection remains open and authenticated so that reauthentication is not required when the session is reused. (WL #12515)

- Connector/NET applications now can use certificates in PEM format to validate SSL connections in addition to the native PFX format (see Tutorial: Configuring SSL with Connector/NET). PEM support applies to both classic MySQL protocol and X Protocol connections. (WL #12494)

Bugs Fixed

- *Document Store:* All methods able to execute a statement were unable to execute the same statement a second time. Now, the values and binding parameters remain available after the method is executed and string parameters are no longer converted to numbers. Both changes enable a follow-on execution to reuse the previous parameters. (Bug #29249857, Bug #29304767)

- An exception was generated when the `MySqlDbType` enumeration was given an explicit value and then passed as a parameter to the `MySqlCommand.Prepare` method. (Bug #28834253, Bug #92912)

- Validation was added to ensure that when a column is of type `TIME` and the value is `00:00:00`, it takes the value instead of setting `NULL`. (Bug #28383726, Bug #91752)

Changes in MySQL Connector/NET 8.0.15 (2019-02-01, General Availability)

Bugs Fixed

- The client library has been modified to initialize the `MySqlBulkLoader` class with the local-infile capability disabled by default (see Using the BulkLoader Class). (Bug #29259767)

Changes in MySQL Connector/NET 8.0.14 (2019-01-21, General Availability)

- Deprecation and Removal Notes
- Functionality Added or Changed

Deprecation and Removal Notes

- The following obsolete (deprecated) members of Connector/NET 8.0 API classes were removed:
  - `Collection.Remove(Object)` method
  - `Collection.Remove(DbDoc)` method
  - `FindStatement.Limit(Int64, Int64)` method
  - `MySqlParameterCollection.Add(String, Object)` method
  - `TableSelectStatement.Limit(Int64, Int64)` method
  - `BaseResult.WarningCount` property
  - `MySqlBaseConnectionStringBuilder.Auth` property
  - `Result.RecordsAffected` property
  - `SqlResult.AutoIncrementValue` property
  - `SqlResult.RecordsAffected` property

  (WL #12031)

Functionality Added or Changed

- The internal method called by the `MySqlX.XDevAPI.Relational.Table.Count`, `MySqlX.XDevAPI.Collection.Count`, and `MySqlX.XDevAPI.Collection<T>.Count` methods were moved to a standardized location within the library. (WL #12518)
- The `auth` connection option (along with aliases `authentication` and `authentication mode`) was removed from the `MySqlBaseConnectionStringBuilder` class. This option now is available for X Protocol connections only. (WL #12031)

Changes in MySQL Connector/NET 8.0.13 (2018-10-22, General Availability)

- Important Changes
- Functionality Added or Changed
- Bugs Fixed

Important Changes

- The default value for the `SslMode` connection option now differs based on the protocol used to make the connection. The `Preferred` mode has been reintroduced in this release (see Options for Both Classic MySQL Protocol and X Protocol). To summarize the default `Sslmode` values in the Connector/NET 8.0 (and 7.0) release series:

  Connector/NET 8.0.13: `Preferred` mode is the default for classic MySQL protocol connections only. `Required` mode is the default for X Protocol connections only (`Preferred` mode is not available for use with X Protocol).

  Connector/NET 8.0.8 to 8.0.12: `Preferred` mode is not supported for any connections. `Required` mode is the default for both classic MySQL protocol and X Protocol connections.

  Connector/NET 7.0.0 to 7.0.7: `Preferred` mode is the default for both classic MySQL protocol and X Protocol connections. (Bug #28687769)

Functionality Added or Changed

- *Document Store:* An incremental improvement was made to the performance of session creation with a connection string. (Bug #28343655)

- Support for EF Core 2.1 was added to Connector/NET 8.0.13 and support for EF Core 2.0 was discontinued in the same connector version. Other versions of Connector/NET continue to support EF Core 2.0 (see Entity Framework Core Support). (WL #12182)

- The `ConnectionTimeout` connection option and property were reimplemented as the `Connect-Timeout` option (and the `ConnectTimeout` property) for X Protocol operations. Some aspects of the timeout behavior were changed (see Options for X Protocol Only).

  The new `ConnectTimeout` property was added to the `MySqlX.XDevAPI.MySqlXConnectionStringBuilder` class and the existing `ConnectionTimeout` property was removed.

  No modifications were made to the existing implementation of the `ConnectionTimeout` option (or property) for classic MySQL operations. (WL #12177)

- Connector/NET now provides connection pooling for applications using the X Protocol. The implementation includes the new static `MySQLX.GetClient` method that takes two parameters: *`connectionData`* (connection string or URI) and *`connectionOptions`* (JSON-formatted string or anonymous object containing the pooling options). Method overloading enables the following type combinations:

  ```
  MySQLX.GetClient(Object, Object)
  MySQLX.GetClient(Object, String)
  MySQLX.GetClient(String, Object)
  MySQLX.GetClient(String, String)
  ```

  `GetClient` returns the new `Client` object, which retrieves an existing and currently unused network connection from the pool, resets it, and uses it. Closing a session marks the underlying connection as unused and returns it to the pool. Connection options are configured as follows:

  ```
  // Connection options of type String
  Client client = MySQLX.GetClient(ConnectionData, "{ \"pooling\": { \"maxSize\":  5, \"queueTimeout\": 5000 }

  // Connection options of type Object (anonymous object)
  Client client = MySQLX.GetClient(ConnectionData, new { pooling = new { maxSize = 5, queueTimeout = 5000 } })
  ```

The `new` keyword must be used twice when the connection options are configured using an anonymous object. Connection options are: `enabled`, `maxSize`, `maxIdleTime`, and `queueTimeout`. For more information, see Connection and Session Concepts. (WL #11841)

Bugs Fixed

- The `CreateCommandBuilder` and `CreateDataAdapter` methods were added to `MySqlClientFactory` class. Thanks to Cédric Luthi for the patch. (Bug #28560189, Bug #92206)
- *Document Store:* Connector-side validation performed when the `Collection.CreateIndex` method was called duplicated some of the checks already made by the server. The connector now only validates that the `indexDefinition` value passed to the method is a valid JSON document with the correct structure. (Bug #28343828)
- *EF Core:* An invalid syntax error was generated when a new property (defined as numeric, has a default value, and is not a primary key) was added to an entity that already contained a primary-key column with the `AUTO_INCREMENT` attribute. This fix validates that the entity property (column) is a primary key first before adding the attribute. (Bug #28293927)
- Connector/NET returned the wrong time zone when the `TIMESTAMP` column was read from a MySQL table. (Bug #28156187)
- *Document Store:* A mixed alphanumeric value such as "1address" when used with the `Set` and `Unset` methods caused the operations to throw an exception. This fix resolves the way mixed values are converted into tokens to generate only one token as an identifier. (Bug #28094094)
- *EF Core:* The implementation of some methods required to scaffold an existing database were incomplete. (Bug #27898343, Bug #90368)
- Attempts to create a new foreign key from within an application resulted in an exception when the key was generated by a server in the MySQL 8.0 release series. (Bug #27715069)
- A variable of type `POINT` when used properly within an application targeting MySQL 8.0 generated an SQL syntax error. (Bug #27715007)
- The implementation of `DbProviderFactory` prevented an application from connecting to MySQL in a generic way. Now, invoking the `CreateDataAdapter` method returns a valid adapter instead of returning the `null` value. (Bug #27292344, Bug #88660)
- The case-sensitive lookup of field ordinals was initialized using case-insensitive comparison logic. This fix removes the original case-sensitive lookup. (Bug #27285641, Bug #88950)
- The `MySql.Data.Types.MySqlGeometry` constructor called with an array of bytes representing an empty geometry collection generated an `ArgumentOutOfRangeException` exception, rather than creating the type as expected. Thanks to Peet Whittaker for the patch. (Bug #26421346, Bug #86974)
- Slow connections made to MySQL were improved by reducing the frequency and scope of operating system details required by the server to establish and maintain a connection. (Bug #22580399, Bug #80030)

Changes in MySQL Connector/NET 8.0.12 (2018-07-27, General Availability)

Known Limitation of This Release

To create a model in Entity Framework (EF) Core v2.0 with scaffolding or migration in this release, use the following alternative procedure:

1. Downgrade to EF Core 1.0 or 1.1 in your project, install the `MySql.Data.EntityFrameworkCore.Design` NuGet package, and then create your model using the scaffolding or migration command.

2. With the model created, revert back to EF Core 2.0 in your project, update the `MySql.Data.EntityFrameworkCore` NuGet package, and then remove the package `MySql.Data.EntityFrameworkCore.Design` from your project.

We are sorry for this temporary inconvenience.

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- Optimistic locking for database-generated fields was improved with the inclusion of the `[ConcurrencyCheck, DatabaseGenerated(DatabaseGeneratedOption.Computed)]` attribute. Thanks to Tony Ohagan for the patch. (Bug #28095165, Bug #91064)

- Members of several classes in the `MySqlX` namespace were added, modified, or removed.

  `MySqlX.XDevAPI.Collection` and `MySqlX.XDevAPI.Collection<T>` classes:

  - `Remove(Object)` method was marked `Obsolete`.
  - `Remove(DbDoc)` method was marked `Obsolete`.
  - `CreateIndex()` method was modified to be a direct-execute method (no longer requires `.Execute()` to execute).

  `MySqlX.XDevAPI.Common.Result` class:

  - `RecordsAffected` property (now obsolete) was replaced with `AffectedItemsCount`.
  - `WarningCount` property (now obsolete) was replaced with `WarningsCount`.

  `MySqlX.XDevAPI.CRUD.FindStatement` class:

  - `GroupBy()` method was added.
  - `Having()` method was added.
  - `Limit(Int64, Int64)` method was marked `Obsolete`.
  - `Offset()` method was added.
  - `Sort()` method was added.
  - `OrderBy()` method was removed.

  `MySqlX.XDevAPI.CRUD.ModifyStatement` class:

  - `ArrayInsert` method was added.
  - `ArrayAppend` method was added.
  - `Unset()` method was modified to accept an array of document paths.

`MySqlX.XDevAPI.CRUD.RemoveStatement` class:

- `Sort()` method was added.
- `OrderBy()` method was removed.

`MySqlX.XDevAPI.Relational.RowResult` class:

- `ColumnCount` property was added.
- `ColumnNames` property was added.

`MySqlX.XDevAPI.Relational.SqlResult` class:

- `AutoIncrementValue` property was marked `Obsolete`.

`MySqlX.XDevAPI.Relational.TableSelectStatement` class:

- `Limit(Int64, Int64)` method was marked `Obsolete`.
- `Offset()` method was added.

`MySqlX.XDevAPI.Session` class:

- `Commit()` method was changed to be a direct-execute method and now it returns `void`.
- `Rollback()` method was changed to be a direct-execute method and now it returns `void`.
- `Uri` property was added.
- `DefaultSchema` property was added.

(Bug #27732098, Bug #27732175, Bug #27732235, WL #11843)

- The `MySqlX.XDevAPI.MySqlXConnectionStringBuilder` class was added to provide connection-string options that apply exclusively to X Protocol connections. In addition, the `Auth`, `SslCa`, and `SslCrl` properties in the `MySql.Data.MySqlClient.MySqlConnectionStringBuilder` class were marked `Obsolete`. (WL #11846)

Bugs Fixed

- *Document Store:* Decimal numbers passed to the `DbDoc` class were not parsed properly when the values included a decimal separator other than a period (`.`) character. (Bug #28112229)
- *Document Store:* The list of members shown with IntelliSense did not match the members provided in the reference documentation. (Bug #27918879, Bug #90615)
- The Entity Framework Core implementation did not render accented characters correctly on bases with different UTF-8 encoding. Thanks to Kleber kleberksms for the patch. (Bug #27818822, Bug #90316)
- The `TreatTinyAsBoolean` connection option was ignored when the `MySqlCommand.Prepare()` method was called. (Bug #27113566, Bug #88472)
- All columns of type `TINYINT(1)` stopped returning the expected Boolean value after the connector encountered a `NULL` value in any column of this type. Thanks to David Warner for the patch. (Bug #22101727, Bug #78917)

Changes in MySQL Connector/NET 8.0.11 (2018-04-19, General Availability)

Functionality Added or Changed

- *Document Store:* Connector/NET now supports the `NOWAIT` and `SKIP_LOCKED` locking options introduced in the MySQL 8.0 release series (see SELECT Statement). The following changes were made to the Connector/NET API:

  - The `LockContention` enumeration (with values `Default=0`, `NoWait=1` and `SkipLocked=2`) was added. The `Default` enumeration member represents the previous behavior of waiting for the row locks to be released.

  - The existing `LockShared()` and `LockExclusive()` method signatures were modified to include the new `LockContention` parameter. Both methods are members of the `MySqlX.XDevAPI.CRUD.FindStatement` and `MySqlX.XDevAPI.Relational.TableSelectStatement` classes.

  - Usage examples:

    ```
    // Default behavior – waits for the row locks to release
    LockShared()
    LockShared(LockContention.Default)

    LockExclusive()
    LockExclusive(LockContention.Default)

    // New – fails if the rows are locked
    LockShared(LockContention.NoWait)
    LockExclusive(LockContention.NoWait)

    // New – succeeds excluding the locked rows from the result
    LockShared(LockContention.SkipLocked)
    LockExclusive(LockContention.SkipLocked)
    ```

    (WL #11307)

- *Document Store:* Previously, when documents without an `_id` attribute were added to a collection, Connector/NET automatically generated IDs for them. Now the server generates the `_id` attribute, unless a document already contains one. The generated IDs resulting from a document-add operation can be obtained using the new `Result.GeneratedIds` property, which returns a list.

  This capability requires a MySQL 8.0 GA server. If the server does not support document ID generation, the document-add operation returns an error indicating that document IDs were missing.

  Incompatibility: The `GeneratedIds` property replaces the `DocumentId` and `DocumentIds` properties, which are now removed. (WL #11421)

- *Document Store:* Support for the SHA256_MEMORY authentication mechanism was added to enable non-PLAIN insecure connections (without SSL) for user accounts with caching_sha2_password, which is the default authentication plugin introduced in the MySQL 8.0 release series. The changes related to this support include:

  - New synonyms for the `auth` connection string option: `authentication` and `authentication mode` (see Options for X Protocol Only).

  - A new authentication mode for the `MySqlAuthenticationMode` enumeration: `SHA256_MEMORY`. In addition, the `Default` member now has a new synonym: `Auto=0`.
  - A new class: `MySql.Data.MySqlClient.Authentication.Sha256MemoryAuthenticationPlugin`.

  (WL #11624)

- Support was added for the new `caching_sha2_password` padding mechanism introduced in the MySQL 8.0 release series. The new padding mechanism is enabled when all of the following conditions apply:
  - The user account is set with the `caching_sha2_password` authentication plugin.
  - SSL is disabled explicitly (`SslMode=none`).
  - The `AllowPublicKeyRetrieval` connection option is enabled (`AllowPublicKeyRetrieval=true`).

  When enabled, the new padding mechanism is used to encode the password during RSA key encryption, which applies the correct padding to match the server. (WL #11618)

Bugs Fixed

- Attempting to open the MySQL Web Configuration Tool, with Connector/NET and MySQL for Visual Studio prerequisites installed properly, displayed an error message instead of opening the tool. (Bug #27457398, Bug #88544)
- Connector/NET could not be installed with NuGet packages from Microsoft Visual Studio 2015. (Bug #27251839, Bug #88838)
- When a decimal column was defined with a scale of zero, such as `DECIMAL(8, 0)`, the value of the `NumericPrecision` field returned by the `MySqlDataReader.GetSchemaTable` method was lower by one. For example, it returned 7 instead of 8 as expected. (Bug #26954812, Bug #88058)
- The data table returned by the `MySqlDataReader.GetSchemaTable` method had an inaccurate value of zero assigned to the `ColumnSize` field for `LONGTEXT` and `LONGBLOB` data types, and also indicated that the `IsLong` field value was `false` when it should have returned `true`. (Bug #26876592, Bug #87876)
- The `MySqlDataReader.GetSchemaTable` method returned different column-size values when used with different character sets. (Bug #26876582, Bug #87868)
- Support for making a secure connection to a server configured to use TLSv1.2 was limited by external factors. (Bug #25689154)
- Connection strings that included TLS/SSL connection parameters in URI type-string format generated an exception instead of making a connection with the X Protocol. (Bug #24510329)
- Attempting to generate an Entity Framework model from a MySQL 5.7 database using either EF5 or EF6 produced an exception that prevented the operation from generating the expected model. (Bug #22173048, Bug #79163)

Changes in MySQL Connector/NET 8.0.10 (2018-01-30, Release Candidate)

- Functionality Added or Changed

- Bugs Fixed

Functionality Added or Changed

- The .NET Core 2.0 implementation now supports the following connection-string options: `AutoEnlist`, `InteractiveSession`, `Logging`, `Replication`, and `UseUsageAdvisor`. (Bug #27297337)
- *Document Store:* In the process of refining the definition of the `MySqlX` namespace to cover the most relevant usage scenarios, the following API components have been removed from the implementation for MySQL Connector/NET:
  - **API components that support session configurations**

    `The MySqlX.XDevAPI.Config` namespace and all members of the namespace.
  - **API components that support views**

    `CreateView()`, `DropView()`, and `ModifyView()` methods from the `MySqlX.XDevAPI.Schema` class.

    `ViewAlgorithm`, `ViewSqlSecurityEnum`, and `ViewCheckOptionEnum` enumerations from the `MySqlX.DataAccess` namespace.

    

    Note

    The `Table.IsView` property remains available for query operations.

  (WL #11306, WL #11362)
- Support for .NET Core 2.0 and .NET Standard 2.0 has been added (.NET Core 1.1 support continues). With .NET Core 2.0, most of the common ADO.NET classes are available for use, such as:
  - `System.Data.DataTable`, `System.Data.DataColumn`, and `System.Data.DataRow`
  - `System.Data.DataSet`
  - `System.Data.Common.DataAdapter`

  (WL #11394)
- Support for Entity Framework Core 2.0 has been added (Entity Framework 1.1 support continues). Currently, the MySQL Connector/NET implementation excludes the following 2.0 features:
  - Modeling: table splitting, owned types, model-level query filters, database scalar function mapping, self-contained type configuration for code first.
  - High performance: DbContext pooling and explicitly compiled queries.
  - Change tracking: attach can track a graph of new and existing entities.
  - Query: improved LINQ translation, group-join improvements, string interpolation in `FromSql` and `ExecuteSqlCommand`, new `EF.Functions.Like()`.
  - Database management: pluralization hook for DbContext scaffolding.
  - Others: only one provider per model, consolidated logging and diagnostics.

  (WL #11395)

- *Document Store:* MySQL Connector/NET now supports setting and releasing named transaction savepoints, which can be assigned a name explicitly or by default using the `savepoint_(uuid)` format. In addition, a transaction can be rolled back to a named savepoint.

  New methods were added to the `MySqlX.XDevAPI.BaseSession` class to implement corresponding SQL statements using the X Protocol:

  - `SetSavepoint()` and `SetSavepoint(name)` correspond to the `SAVEPOINT` statement.
  - `ReleaseSavepoint()` corresponds to the `RELEASE SAVEPOINT` statement.
  - `RollbackTo()` corresponds to the `ROLLBACK TO` statement.

  All errors generated by MySQL when one of the new methods is called will be returned by MySQL Connector/NET. (WL #11135)

- *Document Store:* The `MySqlX.XDevAPI.CRUD.ModifyStatement.Patch` method was added to enable the inclusion of JSON-like objects within `Collection.Modify()` operations that describe the changes to apply to all documents matching the condition. (WL #11133)

- Support for the `caching_sha2_password` authentication plugin through the classic MySQL protocol was added. Support through the X Protocol is limited to secure connections only (`sslmode=required`). Caching SHA-2 pluggable authentication offers faster authentication than basic SHA-256 authentication.

  A new and related connection option, `AllowPublicKeyRetrieval`, was also added. (WL #11081)

- *Document Store:* The `MySqlX.XDevAPI.Collection.CreateIndex` method implementation was modified to enable the inclusion of a JSON document that defines the index to be created. Index-definition details can include the fields affected, data types, and so on. (WL #11131)

Bugs Fixed

- *Document Store:* When the `PLAIN` authentication option was used to make a secure connection, the database name was excluded from the authenticating data and the database value was not set. `PLAIN` authentication is the default option for connections made with TLS or Unix Sockets. (Bug #27098974, Bug #88427)

- Boolean values within a JSON document were improperly stored as strings. (Bug #26837112)

- Invoking the `MySql.Web.Security.MySqlWebSecurity.CreateUserAndAccount` method with valid arguments, including `additionalUserAttributes` as an object with key/value pairs, returned an out-of-range exception. Thanks to Stein Setvik for contributing to the fix. (Bug #25046364)

- When a valid document was passed to the `SetValue` method as a `DbDoc` object, the content within the document was removed. (Bug #24397888)

- The default character set and encoding were not set properly when making a connection to MySQL 5.6 and 5.7 servers configured to use the `utf8` character set. (Bug #23257011)

- After an index was created in MySQL 5.7.12 or higher, an exception prevented the inclusion of additional objects. (Bug #23016623)

- SSL connections made to a single MySQL instance could not be disconnected and created repeatedly without restarting the client application to clear the half-open sockets. (Bug #20393654, Bug #75022)

Changes in MySQL Connector/NET 8.0.9 (2017-09-28, Development Milestone)

- Functionality Added or Changed

- Bugs Fixed

Functionality Added or Changed

- For accuracy, the following Entity Framework 6 items were renamed:
  - NuGet package – `MySql.Data.EntityFramework` (was `MySql.Data.Entity`)
  - Namespace – `MySql.Data.EntityFramework` (was `MySql.Data.Entity`)
  - Assembly – `MySql.Data.EntityFramework.dll` (was `MySql.Data.Entity.EF6.dll`)

  (Bug #26396260)
- *Document Store:* The `SessionConfigManager.Update` method was removed and the `SessionConfigManager.Save` method now always overwrites the data with the given key. For example:

```
SessionConfigManager.Save(
   "mysess",
   "{ \"uri\": \"mysqlx://myuser@localhost/mysess\", \"appdata\": { \"biz\": \"quux\" } }"
);

SessionConfigManager.Save(
   "mysess",
   "{ \"uri\": \"mysqlx://test@localhost/mysess\", \"appdata\": { \"tar\": \"zzzz\" } }"
);
```

  The `mysess.uri` and `mysess.appdata` values set by the first statement are replaced with the new values set by the second statement. (Bug #25829054, Bug #25860579, WL #10954)
- MySQL Connector/NET now supports MySQL servers configured to use utf8mb4 as the default character set. (WL #10562)
- The following methods are available for use with EF Core in asynchronous command and connection operations:
  - `Microsoft.EntityFrameworkCore.DbContext.AddAsync`
  - `Microsoft.EntityFrameworkCore.DbContext.AddRangeAsync`
  - `Microsoft.EntityFrameworkCore.DbContext.FindAsync`
  - `Microsoft.EntityFrameworkCore.DbContext.SaveChangesAsync`
  - `Microsoft.EntityFrameworkCore.Infrastructure.DatabaseFacade.EnsureDeletedAsync`
  - `Microsoft.EntityFrameworkCore.Infrastructure.DatabaseFacade.EnsureCreatedAsync`
  - `Microsoft.EntityFrameworkCore.DbContext.ToListAsync`

  (WL #10615)
- *Document Store:* The following methods execute directly, whereas each method previously required `.execute()` as the final item in the method chain:
  - `BaseSession.DropSchema`
  - `Collection.DropIndex`

  - `Schema.DropCollection`
  - `Schema.DropView`

  In addition, the methods now succeed even if the objects to be dropped do not exist. (WL #10563)

- The `AutoEnlist` and `IncludeSecurityAsserts` connection-string options are not appropriate for use by applications that target .NET Core and now return an error when used. (WL #10564)
- *EF Core:* Support for explicit loading was added. Explicit loading is an object-relational mapper (O/RM) pattern introduced in EF Core 1.1.0, which enables .NET developers to explicitly load related data from the database at a later time. (WL #9768)
- The following connection-string options are not currently supported for use by applications that target .NET Core and now return an error when used:
  - `SharedMemoryName`
  - `IntegratedSecurity`
  - `PipeName`
  - `Logging`
  - `UseUsageAdvisor`
  - `UsePerformanceMonitor`
  - `InteractiveSession`
  - `Replication`

  (WL #11052)
- *Document Store:* To provide safe transactional document and row updates, the following new methods were added:
  - `FindStatement.LockShared`
  - `FindStatement.LockExclusive`
  - `TableSelectStatement.LockShared`
  - `TableSelectStatement.LockExclusive`

  The `LockShared()` and `LockExclusive()` methods can be called any number of times with either the `Collection.Find()` or `Table.Select()` method chains and in any combination. If multiple calls to the methods are made, only the final method is invoked. For additional information about the two types of locking, see Shared and Exclusive Locks. (WL #10948)
- *Document Store:* When creating a new session, multiple hosts can be tried until a successful connection is established. A list of hosts can be given in a connection string or as session creation options, with or without priorities.

```
var mySession = MySQLX.GetSession(
  "mysqlx://dbuser:password@[" +
  "(address=localhost:33060, priority=90)," +
  "(address=192.1.10.10:33060, priority=100)," +
```

```
  "(address=[2001:db8:85a3:8d3:1319:8a2e:370:7348]:33060, priority=30)" +
  "]"
);

var mySession = MySQLX.GetSession(
  "user=dbuser;" +
  "password=dbpassword;" +
  "server=" +
  "(address=192.1.10.10, priority=90)," +
  "(address=server.example.com, priority=100)," +
  "(address=localhost, priority=30);" +
  "port=33060;"
);
```

(WL #10998)

- *Document Store:* The `IN` and `NOT IN` operators have been updated to expand the range of operands that can be evaluated by the `Find()`, `Modify()`, and `Remove()` methods for collections and the `Select()`, `Update()`, and `Delete()` methods for tables. This update provides support for expressions using the following syntax:

  ```
  compExpr ["NOT"] "IN" compExpr
  ```

  The previous syntax used with `IN` and `NOT IN` operators is still valid and it takes precedence over the new syntax when both are present. (WL #10947)

- *Document Store:* Several new direct-execution methods were added to the `Collection` class that operate at a single document level, unlike the other CRUD methods that operate on all documents that match a filter. The new methods are: `ReplaceOne()`, `AddOrReplaceOne()`, `GetOne()`, and `RemoveOne()`. (WL #10949)

- Support for connections using Unix domain socket files was extended to include MySQL servers deployed on Linux hosts.

  X Protocol connection example:

  ```
  "server=/path/to/socket;protocol=unix;user=root;password=mypass;ssl-mode=none"
  ```

  Classic MySQL protocol connection example:

  ```
  "server=/path/to/socket;protocol=unix;user=root;password=mypass"
  ```

  (WL #10201, WL #10613)

- Connections to the MySQL server now can be made using accounts that authenticate with the `sha256_password` plugin. For more information, see SHA-256 Pluggable Authentication.

  In addition, a new connection option was added to extend authentication support for connections made using the X Protocol with either basic or URI connection strings and as an anonymous type. The `auth` connection option enables the `MYSQL41`, `PLAIN`, or `EXTERNAL` authentication mechanism if supported by the server. For a description of the `auth` option, see Options for X Protocol Only. (WL #10595)

Bugs Fixed

- Assemblies within NuGet packages were not fully signed. (Bug #26739307)

- *EF Core:* Some methods in the `DbContext` class were not supported for use with asynchronous operations. (Bug #26448321, Bug #84814)

- *Document Store:* Priority assignment when connecting to the server in client-side failover situations was not supported in earlier versions of the connector by design. Priority-based failover is now available. (Bug #26198794)
- *EF Core:* When attempting to commit a transaction in which the `FirstOrDefaultAsync` method was called, the connector returned `System.InvalidOperationException: Connection must be valid and open to commit transaction` instead of committing the transaction. (Bug #26026972, Bug #86199)
- *Document Store:* Passing in a value of 0 or lower to the `Limit` method now produces a more relevant error message indicating that the argument is out of range. (Bug #24384660)
- *Document Store:* Passing in the NULL value as a parameter to the `DbDoc.SetValue` method resulted in an exception. This fix ensures that NULL is accepted for this method. (Bug #23542093)

Changes in MySQL Connector/NET 8.0.8 (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. For example, MySQL Connector/NET 8.0.12 would be designed to support all features of MySQL server version 8 (or lower). This change makes it easy and intuitive to decide which client version to use for which server version.

MySQL Connector/NET 8.0.8 is the first release to use the new numbering. It is the successor to MySQL Connector/NET 7.0.7.

Functionality Added or Changed

- *Document Store:* The format of document ID values generated when adding documents to a collection has changed. It is still a string of 32 hexadecimal digits based on UUID, but the order of digits was changed to match the requirement of a stable ID prefix. (WL #10202)
- All connections created using MySQL Connector/NET now are encrypted by default. Also, the `Ssl-Enable` connection option has been replaced by `Ssl-Mode`. Permitted `Ssl-Mode` values are `None`, `Required` (the default), `VerifyCA`, and `VerifyFull`.

  With this change, a non-SSL enabled server now requires the `Ssl-Mode` option be set to `None` explicitly in the connection string or the connection will fail. (WL #10559)
- *Document Store:* 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. (WL #10739)
- *Document Store:* The `NodeSession` class has been renamed to `Session` and the `MySQLX.GetNodeSession` method has been renamed to `MySQLX.GetSession`. Also, the `XSession` class has been removed. (WL #10561)
- *Document Store:* When creating a new connection, multiple hosts now can be specified as part of the connection string, which will try each host until a successful connection is established or all elements from the host list have been tried. The following connection-string formats are supported:

```
var mySession = MySQLX.GetSession(
  "mysqlx://dbuser:password@[" +
  "localhost:33060," +
  "192.1.10.10:33060," +
```

```
  "[2001:db8:85a3:8d3:1319:8a2e:370:7348]:33060" +
  "]"
);

var mySession = MySQLX.GetSession(
  "user=dbuser;" +
  "password=dbpassword;" +
  "server=" +
  "192.1.10.10," +
  "server.example.com," +
  "localhost;" +
  "port=33060;"
);
```

(WL #9980)

Bugs Fixed

- *EF Core:* The Database First feature did not support the following data types: `BINARY`, `VARBINARY`, `MEDIUMBLOB`, `LONGBLOB`, `SET`, `DATE`, `TIME`, and `YEAR`. (Bug #25493209)
- *EF Core:* Database First support produced an error when the existing MySQL database included one or more views. (Bug #25493086)
- *EF Core:* Using `System.ComponentModel.DataAnnotations.Schema.TableAttribute` to initialize a new class instance that specified the name of an existing MySQL table produced incorrect mappings of table and column names. (Bug #25394223, Bug #84423)